

The pfSense Documentation

© 2020 Electric Sheep Fencing LLC and Rubicon Communications LLC

Netgate

Sep 10, 2021

CONTENTS

1 Preface 2

2 Introduction 5

3 Releases 13

4 Product Manuals 223

5 Networking Concepts 224

6 Hardware 237

7 Installing and Upgrading 256

8 Configuration 300

9 Backup and Recovery 357

10 Interface Types and Configuration 376

11 User Management and Authentication 399

12 Certificate Management 415

13 Firewall 434

14 Network Address Translation 482

15 Routing 509

16 Bridging 520

17 Virtual LANs (VLANs) 532

18 Multiple WAN Connections 540

19 Virtual Private Networks 559

20 L2TP VPN 642

21 Services 646

22 DHCP 684

23 DNS 686
24 Traffic Shaper 689
25 Captive Portal 706
26 High Availability 727
27 System Monitoring 736
28 Diagnostics 800
29 Packages 823
30 Virtualization 983
31 Wireless 984
32 Cellular Wireless 1005
33 Troubleshooting 1013
34 pfSense Configuration Recipes 1116
35 Menu Guide 1510
36 Glossary of Terms 1515
37 Development 1516
38 References 1544
39 Configuration Recipes 1563
40 Additional Commercial Resources 1564
Index 1565

Thoroughly detailed information and continually updated instructions on how to best operate pfSense® software.

CHAPTER

ONE

PREFACE

1.1 Acknowledgements

This documentation, and the pfSense® project itself, would not be possible without a great team of developers, contributors, customers, and a wonderful community. The project has received code contributions from more than 200 people. Thousands more have done their part supporting the project by helping others on the forum, social media, and other platforms. And even more have contributed by purchasing hardware, support, and services. Our thanks go out to everyone who has done their part to make the project the great success it has become.

1.1.1 pfSense Developers

There are a large number of project and community members, current and in the past, that have contributed to the project, and we thank them all! These following are not comprehensive lists and are presented in no particular order.

The current active pfSense software development team includes:

- Renato Botelho
- Luiz Otavio O Souza
- Jim Pingle
- Jared Dillard
- Steve Beaver
- Matthew Smith

We would also like to recognize several former project members who are no longer active contributors:

- Chris Buechler
- Bill Marquette
- Holger Bauer
- Erik Kristensen
- Seth Mos
- Matthew Grooms

Along with numerous significant community contributors, including:

- Bill Meeks
- Phil Davis
- Anthony (BBCan177)

- Denny Page
- PiBa-NL
- marcelloc
- Stilez

We would also like to thank all FreeBSD developers, specifically, those who have assisted considerably with pfSense project development.

- Max Laier
- Christian S.J. Peron
- Andrew Thompson
- Bjoern A. Zeeb
- George Neville-Neil

1.1.2 Reviewers

The following individuals provided much-needed feedback and insight to help improve the documentation and its accuracy. Listed in alphabetical order by last name.

- Jon Bruce
- Mark Foster
- Bryan Irvine
- Warren Midgley
- Eirik Øverby

1.2 Feedback

The publisher and authors encourage feedback for this documentation and the pfSense® software distribution. Please send suggestions, criticism and/or praise using the feedback forms at the bottom of each page.

For general feedback related to the pfSense project, please post to the forum. Links to these resources can be found at https://www.netgate.com/support/contact-support.html#community-support.

Welcome to *The pfSense Documentation*, written by the pfSense® project team and including contributions from community members.

This set of documents covers topics ranging from the installation process and basic configuration to advanced networking and firewalling using this popular open source firewall and router software distribution.

This is designed to be a friendly guide to common networking and security tasks along with a thorough reference for the capabilities of pfSense software. These documents cover the following topics (and more!):

- An introduction to pfSense software and its features.
- Firewall design and hardware planning.
- Installing and upgrading pfSense software.
- Using the web-based configuration interface.
- Backing up and restoring the firewall configuration.

- Firewalling fundamentals including defining and troubleshooting rules.
- Port forwarding and Network Address Translation (NAT).
- General networking and routing configuration.
- Virtual LANs (VLANs), Multi-WAN, and Bridging.
- Virtual Private Networks using IPsec and OpenVPN.
- Traffic shaping using ALTQ or Limiters.
- Wireless networking configuration.
- Captive Portal setup.
- High Availability using redundant firewalls.
- Various network-related services.
- Firewall monitoring, logging, traffic analysis, sniffing, packet capturing, and troubleshooting.
- Software package and third-party software installations.

There is also a *Menu Guide* with all standard menu choices available in the pfSense software WebGUI.

CHAPTER

TWO

INTRODUCTION

2.1 What does pfSense stand for/mean?

The project ran for months with no name. In fact, the FreeBSD jail that ran the CVS server was called *projectx* until the project was migrated to git several years ago.

Locating an available domain name was the primary difficulty. The project founders, Scott and Chris, ran through numerous possibilities, eventually settling on pfSense® because the firewall would make *sense* of the packet filtering software used, `pf`.

2.2 Why FreeBSD?

Numerous factors came under consideration when choosing a base operating system for the project. This section outlines the primary reasons for selecting FreeBSD.

2.2.1 Wireless Support

Wireless support is a critical feature for many users. In 2004, wireless support in OpenBSD was very limited compared to FreeBSD. OpenBSD did not support drivers or security protocols and offered no plans for their implementation. To this day, FreeBSD surpasses the wireless capabilities of OpenBSD.

2.2.2 Network Performance

Network performance in FreeBSD is significantly better than that of OpenBSD. For small to mid-sized deployments, this generally does not matter; upper scalability is the primary issue in OpenBSD. One pfSense® developer managing several hundred OpenBSD firewalls using `pf` was forced to switch his high load systems to `pf` on FreeBSD to handle the high packets per second rate required by portions of his network. The network performance in OpenBSD has improved since 2004, but limitations still exist.

Multi-processor support for pf in FreeBSD allows for greater scalability and is utilized by pfSense software as seen in this network performance analysis: https://github.com/gvnn3/netperf/blob/master/Documentation/netperf.pdf.

2.2.3 Familiarity and ease of fork

The code for m0n0wall was based on FreeBSD, and pfSense forked from m0n0wall. Changing the base operating system would require prohibitively large modifications and could have introduced limitations from other operating systems, requiring features to be removed or altered.

2.2.4 Alternative Operating System Support

There are no plans to support any other base operating systems at this time.

2.3 Common Deployments

pfSense® software can meet the needs of nearly any type and size of network environment, from a SOHO to datacenter environments. This section outlines the most common deployments.

2.3.1 Perimeter Firewall

The most common deployment of pfSense software is a perimeter firewall. pfSense accommodates networks requiring multiple Internet connections, multiple LAN networks, and multiple DMZ networks. BGP (Border Gateway Protocol), connection redundancy, and load balancing capabilities are configurable as well.

See also:

These advanced features are further described in *Routing* and *Multiple WAN Connections*.

2.3.2 LAN or WAN Router

pfSense software configured as a LAN or WAN router and perimeter firewall is a common deployment in small networks. LAN and WAN routing are separate roles in larger networks.

LAN Router

pfSense software is a proven solution for connecting multiple internal network segments. This is most commonly deployed with VLANs configured with 802.1Q trunking, described more in *Virtual LANs (VLANs)*. Multiple Ethernet interfaces are also used in some environments. High-volume LAN traffic environments with fewer filtering requirements may need layer 3 switches or ASIC-based routers instead.

WAN Router

pfSense is a great solution for Internet Service Providers. It offers all the functionality required by most networks at a much lower price point than other commercial offerings.

2.3.3 Special Purpose Appliances

pfSense can be utilized for less common deployment scenarios as a stand-alone appliance. Examples include: VPN appliance, Sniffer appliance, and DHCP server appliance.

VPN Appliance

pfSense software installed as a separate Virtual Private Network appliance adds VPN capabilities without disrupting the existing firewall infrastructure, and includes multiple VPN protocols.

Sniffer Appliance

pfSense offers a web interface for the `tcpdump` packet analyzer. The captured `.cap` files are downloaded and analyzed in Wireshark.

See also:

For more information on using the packet capture features of pfSense, see *Packet Capturing*.

DHCP Server Appliance

pfSense software can be deployed strictly as a Dynamic Host Configuration Protocol server, however, there are limitations of the pfSense GUI for advanced configuration of the ISC DHCP daemon.

See also:

For more information on configuring the DHCP service on pfSense, see *DHCP*.

2.4 Interface Naming Terminology

All interfaces on a pfSense® router/firewall can be assigned any name desired, but they all start with default names: WAN, LAN, and OPT.

2.4.1 WAN

Short for *Wide Area Network*, WAN is the untrusted public network outside of the firewall. In other words, the WAN interface is the firewall's connection to the Internet or other upstream network. In a multi-WAN deployment, WAN is the first or primary Internet connection.

At a minimum, the firewall must have one interface, and that is WAN.

2.4.2 LAN

Short for *Local Area Network*, LAN is commonly the private side of a firewall. It typically utilizes a *private IP address* scheme for local clients. In small deployments, LAN is typically the only internal interface.

2.4.3 OPT

OPT or *Optional* interfaces refer to any additional interfaces other than WAN and LAN. OPT interfaces can be additional LAN segments, WAN connections, DMZ segments, interconnections to other private networks, and so on.

2.4.4 DMZ

Short for the military term *demilitarized zone*, DMZ refers to the buffer between a protected area and a war zone. In networking, it is an area where public servers are reachable from the Internet via the WAN but isolated from the LAN. The DMZ keeps the systems in other segments from being endangered if the network is compromised, while also protecting hosts in the DMZ from other local segments and the Internet in general.

Warning: Some companies misuse the term "DMZ" in their firewall products as a reference to 1:1 NAT on the WAN IP address which exposes a host on the LAN. For more information, see *1:1 NAT on the WAN IP, aka "DMZ" on Linksys*.

2.4.5 FreeBSD interface naming

The name of a FreeBSD interface starts with the name of its network driver. It is then followed by a number starting at `0` that increases incrementally by one for each additional interface sharing that driver. For example, a common driver used by Intel gigabit network interface cards is `igb`. The first such card in a system will be `igb0`, the second is *igb1*, and so on. Other common driver names include `cxl` (Chelsio 10G), `em` (Also Intel 1G), `ix` (Intel 10G), `bge` (various Broadcom chipsets), amongst numerous others. If a system mixes an Intel card and a Chelsio card, the interfaces will be `igb0` and `cxl0` respectively.

See also:

Interface assignments and naming are further covered in *Installing and Upgrading*.

2.5 Finding Information and Getting Help

This section offers guidance on finding information in this documentation, on pfSense® software in general, as well as providing further resources.

2.5.1 Finding Information

The search function on the documentation is the easiest way to find information on a specific topic. The most common features and deployments of pfSense are covered in this documentation. When reading the HTML version of the documentation, the search function is in the upper left of the page. When reading an eBook style copy, consult the documentation for the book reader software for information on how to search.

There is a wealth of additional information and user experiences available on the various netgate.com websites. The best way to search the sites is a Google search appending `site:netgate.com` to the query. This will search the website, forum, documentation, etc. which are all official sources of information.

2.5.2 Getting Help

A help icon is available on almost every page,  , and links to the associated page in documentation.

Netgate offers several other ways to get help with pfSense software, including the Netgate Forum, this documentation, and the pfSense subreddit. More information can be found on the Netgate website at Obtaining Support Many of these links are reachable from the the Help menu in the GUI.

2.6 Comparison to Commercial Alternatives

The question of security and support vs. commercial alternatives comes up from time to time. The history of this project since its inception in 2004 proves we're as secure as any, and better than many, commercial alternatives. The experiences of our customers proves not only can we match the service of any commercial firewall vendor, we exceed it. This page serves to debunk the common myths when comparing to commercial alternatives.

2.6.1 "Hardware" firewalls are better myth

Commercial firewall companies' marketing departments have done a fine job ingraining the myth of "hardware firewalls" into some people's minds. The reality is there is no such thing as a "hardware firewall." All firewalls are hardware that runs software. Most commercial firewalls are based on BSD (same as pfSense®) or Linux. Numerous commercial firewalls run many of the same underlying software programs that pfSense uses. Many commercial alternatives run on x86 hardware that's no different from what people use for pfSense. In fact many people have loaded pfSense on hardware that used to run their commercial firewall, including Watchguard, Nortel, Barracuda and more.

2.6.2 Open source is insecure myth

Some people are of the mindset that because the source is open, it's insecure because everyone can see how it works. Anyone who has paid any attention to security over the past 20 years knows the absurdity of that statement. No software relies on the obscurity of source code for security. If there was any truth in that, Microsoft Windows would be the most secure OS ever created, when the reality is all of the open source operating systems (all the BSDs and Linux) have security track records that are worlds better than Windows'. History proves the same applies to any software. Internet Explorer is continually hit with major security holes that many times take weeks to patch while they're being exploited in the wild, while open source browsers Firefox, Chrome and others have had significantly better security track records.

The widespread UPnP vulnerabilities announced in 2013 affecting over 300 commercial products is another good example. The vendors of hundreds of commercial products made extremely basic security mistakes, shipping with absurdly insecure defaults, and shipping outdated software. That's never been an issue with pfSense. That's just one example of where we've done a better job than many commercial vendors.

2.6.3 Commercial alternatives have better support myth

With some open source projects, it's true that a user is stuck if they need help. Netgate offers commercial support for pfSense software, Netgate TAC, that rivals anything other commercial vendor offers.

2.7 Can pfSense meet regulatory requirements

Prospective pfSense® users commonly inquire about the ability to meet security requirements applicable to their specific environments. Some of those include PCI, SOX, GLBA, HIPAA, amongst numerous other similar regulations for publicly traded companies, financial institutions, healthcare institutions, and others.

There are numerous companies in many regulated industries using pfSense that pass their audits with no problems, including all of the aforementioned regulations/standards amongst others. However it's important to keep in mind that a firewall is a small portion of the security infrastructure, and those regulations are more about policies, procedures, and configuration than the actual products being used.

So yes, pfSense *can* meet regulatory requirements, but that is dependent on configuration, policies, procedures, amongst other things - there is no compliance silver bullet. There may be circumstances specific to one company that make another product a better fit for compliance (or other) reasons, but that's true of all commercial and open source solutions, there is no one product that is a perfect fit for everyone.

2.8 Can I sell pfSense

Many consulting companies offer pfSense® solutions to their customers. A business or individual can load pfSense for themselves, friends, relatives, employers, and, yes, even customers, so long as the Trademark Guidelines and Apache 2.0 license requirements as detailed on the website are obeyed by all parties involved.

What can not be offered is a commercial redistribution of pfSense® software, for example the guidelines do not permit someone to offer "Installation of pfSense® software" as a service or to sell a device pre-loaded with pfSense® software to customers without the prior express written permission of ESF pursuant to the trademark policy.

Example 1: A consultant may offer firewall services (e.g. "Fred's Firewalls"), without mentioning pfSense® software or using the logo in their advertising, marketing material, and so on. They can install pfSense® software and manage it for their customers.

Example 2: Fred's Firewalls may make a customized distribution pfSense® software with their own name and logo used in place of the pfSense marks. They can use the pfSense marks to truthfully describe the origin of the software, such as "Fred's Firewall software is derived from the pfSense CE source code." Even though Fred's Firewall is based on pfSense® software, it can **not** be referred to as "pfSense® software" since it has been modified.

Example 3: Fred's Firewalls may sell their customized firewall distribution pre-loaded on systems to customers, so long as the relationship to pfSense is clearly stated.

The Apache 2.0 license only applies to the software and **not** the pfSense name and logo, which are trademarks and may not be used without a license. Reading and understanding the trademark policy document is required before one considers selling pfSense Software.

2.8.1 Contributing Back to the Project

We ask anyone profiting by using pfSense software to contribute to the project in some fashion. Ideally with the level of contributions from a business or individual corresponding to the amount of financial gain received from use of pfSense software. Many paths exist for resellers and consultants to contribute. For the long term success of the project this support is critically important.

1. Purchase hardware and merchandise from the Netgate Store.
2. Become a Netgate Partner to resell Netgate hardware pre-loaded with pfSense software.
3. Development contributions - Dedicate a portion of internal developers' time to open source development.
4. Help with support and documentation - Assisting users on the forum and mailing list, or contributing documentation changes, aides the overall project.
5. Support subscription via Netgate TAC Having direct access to our team for any questions or deployment assistance helps ensure success.

2.8.2 Using the pfSense Name and Logo

The "pfSense" name and logo are trademarks of Electric Sheep Fencing, LLC.

The pfSense software source code is open source and covered by the Apache 2.0 license. That license only covers the source code and **not** our name and trademarks, which have restricted usage.

We think it is great that people want to promote and support the pfSense project. At the same time, we also need to verify that what is referred to as "pfSense" is a genuine instance of pfSense software and **not** modified in any way.

- The pfSense name and logo **MAY NOT** be used physically on a hardware device.
  - For example: A sticker, badge, etching, or similar rendering of the pfSense name or logo is NOT allowed.
- The pfSense logo **MAY NOT** be used on marketing materials or in other ways without a license, including references on websites.
- The pfSense name **MAY** be used to describe the case that a product is based on a pfSense distribution, but the designated product name may *not* include pfSense or a derivative. Basically stating facts regarding product origin is acceptable. Anything that implies that a product is endorsed by or made by ESF or the pfSense project is *not* allowed.
  - Some examples:
    - "Blahsoft Fireblah based on pfSense software" – Acceptable
    - "Blahsoft pfSense Firewall" – NOT Allowed
- **ONLY** an **UNMODIFIED** version of pfSense software can still be called "pfSense software".
  - If the source code has been changed, compiled/rebuilt separately, included extra file installations such as themes or add-on scripts, or any other customizations, it can **not** be called "pfSense software", it must be called something else.
  - Trademark protection aside, this requirement preserves the integrity and reputation of the pfSense project. It also prevents unverified changes that may be questionably implemented from being attributed to pfSense.
- If a pfSense distribution is modified, the resulting software **CANNOT** be called "pfSense" or anything *similar*. The new name must be distinct from pfSense. Trademark law does not allow use of names or trademarks that are confusingly similar to the pfSense Marks. This means, among other things, that law forbids using a variation of the pfSense Marks, their phonetic equivalents, mimicry, wordplay, or abbreviation with respect to similar or related projects, products, or services (for example, "pfSense Lifestyle," "PFsense Community," "pf-Sense Sensibility," "pfSensor", etc., **all** infringe on ESF's rights).

  - Examples:
    - "pfSomething", or "somethingSense" – INFRINGING references
    - "ExampleWall", "FireWidget" – NON-Infringing references
- The "pfSense" name **MAY NOT** be used in a company name or similar. A company **CANNOT** be named "pfSense Support, Ltd" or "pfSense Experts, LLC", or use it in a domain name or subdomain reference. However, the company can state support for pfSense software, offer training for pfSense software, etc.
- There **MUST** be a distinction between a company name and pfSense or Electric Sheep Fencing, LLC. No relationship or endorsement can be stated or implied between the two companies, unless we have explicitly licensed and agreed to such a statement.

The pfSense® Project is a free open source customized distribution of FreeBSD tailored for use as a firewall and router entirely managed by an easy-to-use web interface. This web interface is known as the web-based GUI configurator, or WebGUI for short. No FreeBSD knowledge is required to deploy and use pfSense software. In fact, the majority of users have never used FreeBSD outside of pfSense software. In addition to being a powerful, flexible firewalling and routing platform, pfSense software includes a long list of related features. The pfSense package system allows further expandability without adding bloat and potential security vulnerabilities to the base distribution. pfSense is a popular project with millions of downloads since its inception and hundreds of thousands of active installations. It has been proven successful in countless installations ranging from single computer protection in small home networks to thousands of network devices in large corporations, universities and other organizations.

To download the latest version, see previous versions, or to upgrade follow the guides located on the pfSense downloads page.

2.9 Project Inception

This project was founded in 2004 by Chris Buechler and Scott Ullrich. Chris contributed to m0n0wall for some time prior and found it to be a great solution. Although thrilled with the project, many users longed for more capabilities than those accommodated by a project strictly focused towards embedded devices with their limited hardware resources. Enter pfSense. In 2004, there were numerous embedded solutions with 64 MB RAM that couldn't accommodate the desired feature set of pfSense, thus pfSense expanded to work on more capable PC and server type hardware.

CHAPTER

THREE

RELEASES

This section contains information about past and present release of pfSense® software. This includes release notes and detailed version information.

3.1 General Release Information

3.1.1 Versions of pfSense software and FreeBSD

The tables in this document contain detailed information on pfSense® software releases.

The versions are grouped up by major/minor changes so they are easier to locate, and the most recent versions are listed first.

pfSense Plus

21.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
21.05.1		2021-08-05	21.7	12.2-STABLE@424f6363927	plus-RELENG_21_05_1
21.05		2021-06-02	21.7	12.2-STABLE@424f6363927	plus-RELENG_21_05
21.02.2		2021-04-13	21.5	12.2-STABLE@f4d0bc6aa6b	plus-RELENG_21_02_2
21.02-p1		2021-02-25	21.4	12.2-STABLE@f4d0bc6aa6b	plus-RELENG_21_02
21.02		2021-02-17	21.4	12.2-STABLE@f4d0bc6aa6b	plus-RELENG_21_02

pfSense CE

2.5.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.5.2	✔	2021-07-07	21.7	12.2-STABLE@f4d0bc6aa6b	RELENG_2_5_2
2.5.1	✔	2021-04-13	21.5	12.2-STABLE@f4d0bc6aa6b	RELENG_2_5_1
2.5.0	✔	2021-02-17	21.4	12.2-STABLE@f4d0bc6aa6b	RELENG_2_5_0

2.4.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.4.5-p1	✔	2020-06-09	19.1	11.3-STABLE@r357046	RELENG_2_4_5
2.4.5	✘	2020-03-26	19.1	11.3-STABLE@r357046	RELENG_2_4_5
2.4.4-p3	✘	2019-05-20	19.1	11.2-RELEASE-p10	RELENG_2_4_4
2.4.4-p2	✘	2019-01-07	18.9	11.2-RELEASE-p4	RELENG_2_4_4
2.4.4-p1	✘	2018-12-03	18.9	11.2-RELEASE-p4	RELENG_2_4_4
2.4.4	✘	2018-09-24	18.8	11.2-RELEASE-p3	RELENG_2_4_4
2.4.3-p1	✘	2018-05-14	18.0	11.1-RELEASE-p10	RELENG_2_4_3
2.4.3	✘	2018-03-29	17.9	11.1-RELEASE-p7	RELENG_2_4_3
2.4.2-p1	✘	2017-12-14	17.3	11.1-RELEASE-p6	RELENG_2_4_2
2.4.2	✘	2017-11-20	17.3	11.1-RELEASE-p4	RELENG_2_4_2
2.4.1	✘	2017-10-24	17.3	11.1-RELEASE-p2	RELENG_2_4_1
2.4	✘	2017-10-12	17.0	11.1-RELEASE-p1	RELENG_2_4_0

2.3.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.3.5-p2	✖	2018-05-14	15.8	10.3-RELEASE-p26	RELENG_2_3_5
2.3.5-p1	✖	2017-12-14	15.8	10.3-RELEASE-p26	RELENG_2_3_5
2.3.5	✖	2017-10-31	15.8	10.3-RELEASE-p20	RELENG_2_3_5
2.3.4-p1	✖	2017-07-20	15.8	10.3-RELEASE-p19	RELENG_2_3_4
2.3.4	✖	2017-05-04	15.8	10.3-RELEASE-p19	RELENG_2_3_4
2.3.3-p1	✖	2017-03-09	15.8	10.3-RELEASE-p17	RELENG_2_3_3
2.3.3	✖	2017-02-20	15.8	10.3-RELEASE-p16	RELENG_2_3_3
2.3.2	✖	2016-07-19	15.5	10.3-RELEASE-p5	RELENG_2_3_2
2.3.1	✖	2016-05-18	15.4	10.3-RELEASE-p3	RELENG_2_3_1
2.3	✖	2016-04-12	15.0	10.3-RELEASE	RELENG_2_3_0

2.2.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.2.6	✖	2015-12-21	12.0	10.1-RELEASE-p25	RELENG_2_2
2.2.5	✖	2015-11-05	12.0	10.1-RELEASE-p24	RELENG_2_2
2.2.4	✖	2015-07-26	11.9	10.1-RELEASE-p15	RELENG_2_2
2.2.3	✖	2015-06-24	11.7	10.1-RELEASE-p13	RELENG_2_2
2.2.2	✖	2015-04-15	11.7	10.1-RELEASE-p9	RELENG_2_2
2.2.1	✖	2015-03-17	11.7	10.1-RELEASE-p6	RELENG_2_2
2.2	✖	2015-01-23	11.6	10.1-RELEASE-p4	RELENG_2_2

2.1.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.1.5	✖	2014-08-27	10.1	8.3-RELEASE-p16	RELENG_2_1
2.1.4	✖	2014-06-25	10.1	8.3-RELEASE-p16	RELENG_2_1
2.1.3	✖	2014-05-02	10.1	8.3-RELEASE-p16	RELENG_2_1
2.1.2	✖	2014-04-10	10.1	8.3-RELEASE-p14	RELENG_2_1
2.1.1	✖	2014-04-04	10.1	8.3-RELEASE-p14	RELENG_2_1
2.1	✖	2013-09-15	9.8	8.3-RELEASE-p11	RELENG_2_1

2.0.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
2.0.3	✖	2013-04-15	8.0	8.1-RELEASE-p13	RELENG_2_0
2.0.2	✖	2012-12-21	8.0	8.1-RELEASE-p13	RELENG_2_0
2.0.1	✖	2011-12-20	8.0	8.1-RELEASE-p6	RELENG_2_0
2.0	✖	2011-09-17	8.0	8.1-RELEASE-p4	RELENG_2_0

1.2.x

Version	Support	Released	Config Rev	FreeBSD Version	Branch
1.2.3	✖	2009-12-10	3.0	7.2-RELEASE-p5	RELENG_1_2
1.2.2	✖	2009-01-09	3.0	7.0-RELEASE-p8	RELENG_1_2
1.2.1	✖	2008-12-26	3.0	7.0-RELEASE-p7	RELENG_1_2
1.2	✖	2008-02-25	3.0	6.2-RELEASE-p11	RELENG_1_2

Legend

Version The pfSense software version number, and when possible, the version number links to the release notes detailing what was changed in that particular release.

Support The support status.

 Current supported release

 Previous unsupported release

 Future release

TBD To Be Determined, not yet known.

Released The date a specific version of pfSense was released to the public.

Config Rev The internal `config.xml` revision number, which indicates changes to the configuration format that may make a configuration file incompatible with older versions.

FreeBSD Version Each version of pfSense is based on a specific version of FreeBSD. The underlying FreeBSD version is listed for each corresponding version of pfSense.

Branch A link to the pfSense software source code branch used to build a specific release.

3.2 Current/Upcoming Supported Releases

3.2.1 21.09 New Features and Changes

This is a regularly scheduled software release of pfSense Plus software including new features, additional hardware support, and bug fixes.

> **Warning:** When upgrading to pfSense Plus 21.09 and later versions, the `pfSense-upgrade` process will forcefully reinstall all operating system packages and add-on packages to ensure a consistent state and package set. This may increase the time the upgrade will take to download and install.

General

- This release contains several significant changes to IPsec for stability and performance. Read the IPsec section of this document carefully.

  > **Warning:** IPsec VTI interface names have changed in this release. Configurations will be updated automatically where possible to use the new names. If any third party software configurations or other manual changes referenced the old IPsec VTI interface names directly (e.g. `ipsecNNNN`) they must be updated to the new format.

- ZFS is now the default filesystem for new installation on most hardware. Firewalls with UFS will still run UFS after upgrading as installations cannot be changed in-place from UFS to ZFS. To migrate from UFS to ZFS, use an installation image to wipe and reload the firewall.

- **Log Compression** for rotation of System Logs is now disabled by default for new ZFS installations as ZFS performs its own compression.

Tip: The best practice is to disable **Log Compression** for rotation of System Logs manually for not only existing ZFS installations, but also for any system with slower CPUs. This setting can be changed under **Status > System Logs** on the **Settings** tab.

- The default password hash format in the User Manager has been changed from Blowfish to SHA-512. New users created in the User Manager will have their password stored as a SHA-512 hash. Existing user passwords will be changed to SHA-512 next time their password is changed.

Note: User Manager passwords are only stored as a hash, thus existing users cannot be automatically changed to the new format. To convert a user password from an older hash format, change the password for the user in the User Manager.

---

pfSense Plus

Aliases / Tables

- Fixed: Error loading rules when URL Table Ports content is empty #4893
- Fixed: Mixed use of aliases in a port range produces unloadable ruleset #11818
- Fixed: Unable to create nested URL aliases #11863
- Fixed: Creating or editing aliases fails with multiple hosts separated by spaces #12124
- Fixed: When attempting to delete an in-use alias, input validation only prints the first item using the alias in the error message #12177

Authentication

- Changed: Use SHA-512 for user password hashes #10298

Backup / Restore

- Fixed: Output from reboot process is printed on Backup & Restore page when restoring a configuration file #11909
- Fixed: Custom value for AutoConfigBackup schedule Hours is not shown when loading the settings page #11946
- Fixed: Viewing an AutoConfigBackup entry takes approximately 60 seconds to completely load #12247

Build / Release

- Changed: Remove deprecated `libzmq` code and references #12060

CARP

- Fixed: Cannot enter persistent CARP maintenance mode when CARP is disabled #11727
- Fixed: When a CARP VIP VHID change is synchronized to a secondary node, the CARP VIP is removed from the interface and the old VHIDs remain active #12202
- Fixed: Changing VHID on CARP VIP does not update VHID of related IP Alias VIPs #12227

Captive Portal

- Fixed: Vouchers may expire too early when using RAM disks #11894
- Fixed: Incorrect variable substitution in captive portal error page #11902
- Fixed: Clicking "logout" on portal page does not function when logout popup is disabled #12138

Certificates

- Fixed: Certificate Revocation tab does not list active users of CRL entries #11831
- Fixed: Certificate manager reports CA as in use by an LDAP server when LDAP is not configured for TLS #11922
- Fixed: Certificate Manager performs redundant escaping of special characters in certificate DN fields #12034
- Fixed: Certificate Manager shows incorrect DN for imported entries with UTF-8 encoding #12041

Console Menu

- Fixed: Cannot configure WAN IP address with `/32` CIDR mask via console menu #11581

DHCP (IPv4)

- Added: Support for UEFI HTTP Boot option in DHCPv4 Server #11659
- Fixed: DHCPv4 server configuration does not include ARM TFTP filenames #11905
- Fixed: ARM 32/64 network boot options are not parsed on Static DHCP Mapping page #12216

DHCP (IPv6)

- Fixed: DHCPv6 Server should not offer configuration options for unsupported PPPoE Server interfaces #12277

DHCP Relay

- Fixed: PHP error if no DHCPv6 Relay interfaces are selected #11969

DNS Resolver

- Fixed: Unbound crashes with signal 11 when reloading #11316
- Fixed: Unbound fails to start if its configuration references a python script which does not exist #12274

Dashboard

- Fixed: System Information widget unnecessarily polls data for hidden items #12241
- Fixed: IPsec widget generates errors if no tunnels are defined #12337
- Added: Disks dashboard widget to replace Disk Usage section of System Information widget #12349

Diagnostics

- Fixed: State table content on `diag_dump_states.php` does not sort properly #11852
- Changed: Hide "Reboot and run a filesystem check" for ZFS systems #11983
- Fixed: "GoTo line #" function does not work on `diag_edit.php` #12050
- Fixed: Sanitize WireGuard private and pre-shared keys in status output #12256
- Added: Include firewall rules from packages which failed to load in status output #12269

Dynamic DNS

- Added: Option to set interval of forced Dynamic DNS updates #9092
- Added: Support DNS Made Easy authentication without a username #9341
- Fixed: RFC 2136 Dynamic DNS client uses IPv6 alias VIP instead of Track IPv6 address for AAAA records #11816
- Added: New Dynamic DNS Provider: Strato #11978
- Fixed: Dynamic DNS cache expiration time check calculation method may cause update to happen on the wrong day #12007
- Fixed: NoIP.com incorrectly encodes Dynamic DNS update credentials #12021
- Added: New Dynamic DNS Provider: deSEC #12086
- Added: Support Check IP services which return bare IP address values #12194
- Fixed: Yandex Dynamic DNS client does not set the `PddToken` value #12331

FreeBSD

- Fixed: Duplicate `comconsole_port` lines in `/boot/loader.conf` #11653
- Changed: Upgrade to `pkg` 1.17.x #12171

Gateways

- Fixed: Default IPv4 gateway may be set to IPv6 gateway value in certain cases #12282

High Availability

- Fixed: Incorrect RADVD log message on HA event #11966

IPsec

- Fixed: Disconnected IPsec phase 2 entries are not shown in IPsec status #6275
- Fixed: UDP fragments received over IPsec tunnel are not properly reassembled and forwarded #7801
- Fixed: EAP-RADIUS Mobile IPsec clients with RADIUS-assigned addresses do not get additional configuration attributes #11447
- Fixed: Incorrect phase 2 entry removed when deleting multiple items consecutively #11552
- Fixed: strongSwan configuration contains incorrect structure for mobile pool DNS records #11891
- Fixed: IPsec status tunnel descriptions are incorrect #11910
- Changed: PC/SC Smart Card Daemon `pcscd` running on all devices at all times, should be optional #11933
- Fixed: IPsec status fails when many tunnels are connected #11951
- Fixed: Mobile IPsec advanced RADIUS parameters do not allow numeric values with a decimal point #11967
- Fixed: Mobile IPsec NAT/BINAT entries missing from firewall rules #12023
- Fixed: Applying IPsec settings for many tunnels is slow or times out #12026
- Fixed: Gateway alarm always triggers IPsec restart #12039
- Changed: Improve IPsec identifier settings #12044
- Fixed: IPsec status IKE disconnect button drops all connections for the IKE ID, not a specific IKE SA ID #12052
- Fixed: Tunnels with conflicting REQID values can lead to multiple identical Child SA entries #12155
- Added: IPsec keep alive option to initiate phase 2 without using ICMP #12169
- Added: Add connect/disconnect buttons to IPsec dashboard widget #12181
- Fixed: IPsec status shows connect buttons while tunnel is connecting #12189
- Fixed: IPsec writes CRL files when tunnel does not use certificates #12195
- Fixed: IPsec settings fail to apply when a remote gateway is set to an FQDN and there are no DNS servers available #12196
- Fixed: Mobile IPsec phase 1 should not display “Gateway duplicates” option #12197
- Fixed: Disabling an IPsec phase 1 entry does not disable related phase 2 entries #12198

- Fixed: Disabled IPsec VTI interfaces are always created #12212
- Fixed: IPsec bypass rules display help text under each entry #12236
- Fixed: IPsec phase 1 entry with `0.0.0.0` as its remote gateway does not receive correct automatic firewall rules #12262
- Changed: Update "IPsec Filter Mode" option values and help text to reflect that VTI mode also helps transport mode (e.g. GRE) #12289
- Fixed: IPsec manual initiation and termination should use a timeout value or forced actions #12298
- Fixed: IPsec tunnels using a gateway group do not get reloaded in some cases #12315
- Fixed: IPsec Phase 2 entry incorrectly orders proposals in AH mode #12323
- Fixed: Hash algorithm GUI options are disabled after switching a phase 2 entry to AH mode #12324
- Fixed: IPsec VTI interface remote endpoint is not resolved the correct way #12328

IPv6 Router Advertisements (RADVD)

- Fixed: "Default preferred lifetime" router advertisement validation check uses incorrect variable #12159
- Fixed: IPv6 RA DNSSL lifetime is too short, not compliant with RFC 8106 #12173
- Fixed: Default IPv6 router advertisement intervals and lifetime are too low #12280

Interfaces

- Fixed: GRE and GIF tunnels on dynamic IPv6 interface are not brought up during boot #6507
- Fixed: Interface column empty in list of GIF tunnels when using IP Alias on CARP VIP as Interface #11337
- Fixed: QinQ using OpenVPN `ovpn` interface as a parent is not configured at boot time #11662
- Fixed: VLAN and QinQ edit pages allows selecting incompatible OpenVPN `tun` interfaces #11675
- Fixed: Advanced DHCP client configuration "Protocol timing" help text is in the wrong location #11926
- Added: VLAN list sorting #11968
- Fixed: Boot messages contain entries about configuring LAGG/VLAN/QinQ interfaces even when no entries of those types are configured #12002
- Fixed: Input validation incorrectly rejects a second IPv4-only GRE tunnel #12049
- Fixed: Interface assignment mismatch is not detected if VLAN-only parent interface is removed #12170
- Fixed: IPv6 DNS servers from dynamic sources are not listed on `status_interfaces.php` #12252
- Fixed: IPv6 gateway for an interface is not shown on `status_interfaces.php` if the interface does not also have an IPv4 gateway #12253

L2TP

- Fixed: Kernel panic during L2TP retransmit #9058
- Fixed: FQDN L2TP server address is only resolved at boot #12072

Logging

- Fixed: Logging configuration added by a package is not removed on uninstall #11846
- Fixed: Remote log server input validation allows invalid values #12000
- Added: Disable log compression on new installations when `/var/log` is a ZFS dataset with compression enabled #12011
- Changed: Improve log settings help text for file size, compression, and retention count #12012
- Added: Create a log entry when a configuration change occurs #12118

NTPD

- Added: Support SHA-256 hash NTP authentication #12213

Notifications

- Added: Option to suppress expiration notifications for revoked certificates #12109

OpenVPN

- Added: Support aliases in OpenVPN local/remote/tunnel network fields #2668
- Changed: Set `explicit-exit-notify` option by default for new OpenVPN server instances #11684
- Fixed: OpenVPN client certificate validation with OCSP always fails #11829
- Added: Option to validate OpenVPN peer TLS certificate key usage #11865
- Added: Log external IP address of OpenVPN clients on connect and disconnect #11935
- Fixed: DNS Resolver does not add PTR record for OpenVPN clients #11938
- Fixed: OpenVPN IPv6 tunnel network is not validated properly #11999
- Fixed: OpenVPN RADIUS-based firewall rules use incorrect port ranges #12020
- Fixed: Incorrect OpenVPN Client Export help link #12022
- Fixed: OpenVPN RADIUS-based firewall rules do not use expected value for RADIUS-assigned IP addresses #12076
- Fixed: Prevent using OpenVPN "Exit Notify" option with point-to-point modes #12102
- Fixed: OpenVPN Wizard configuration missing recently added default values #12172
- Fixed: OpenVPN does not clean up previous CA and CRL files #12192
- Changed: Move "Description" option on OpenVPN server and client pages to top of the page, show internal instance ID #12218

- Fixed: Prevent using OpenVPN "Inactive" option with point-to-point modes #12219
- Fixed: Configuration files are not deleted after disabling an OpenVPN instance #12223
- Fixed: OpenVPN page allows to delete/disable instance with an assigned interface #12224
- Fixed: OpenVPN status incorrect for TAP servers without a defined tunnel network #12232
- Fixed: OpenVPN client connect/disconnect scripts are not used in Remote Access (SSL/TLS) mode #12238

Operating System

- Changed: Ensure `/usr/local/sbin/` scripts use full path to executable files #11985
- Fixed: Update NGINX to address CVE-2021-23017 #12061
- Added: Suppress kernel messages for `lo0` configuration during boot #12094
- Changed: Convert RAM disks to `tmpfs` #12145

PHP Interpreter

- Fixed: PHP exits with signal 11 on SG-3100 when calling PCRE functions #11466

PPP Interfaces

- Fixed: PPP interfaces lose the description field in `ifconfig` output when restarted #11959

Package System

- Fixed: Package `<plugins>` and `<tabs>` content missing from configuration in some cases #11290
- Fixed: Packages are not automatically reinstalled when restoring configuration using the installer #12105

RRD Graphs

- Added: Graph for hardware temperature readings #9297

Rules / NAT

- Added: IPv6 support in `easyrule` CLI script #11439
- Fixed: NAT rule overlap detection is inconsistent #11734
- Fixed: Input validation not working for 1:1 NAT entries using an alias as a destination #11923
- Fixed: `easyrule` script does not function properly #12151
- Fixed: IPv6 policy routing does not work if an IPsec tunnel phase 2 remote network is configured for `::/0` #12164
- Fixed: 1:1 NAT rule with internal IP address of "Any" results in an invalid firewall rule #12168
- Fixed: Firewall rule tabs load slowly when many rules on the tab utilize gateways #12174

- Fixed: VIP network addresses are not expanded on Port Forward rules #12233
- Fixed: Duplicating a Port Forward does not copy "Filter Rule Association" values of "None" or "Pass" #12272

Services

- Fixed: System attempts to stop inactive services at shutdown #12001
- Fixed: System attempts to start inactive services at boot #12038

Traffic Shaper (ALTQ)

- Fixed: Panic when using CBQ traffic shaping #11470

UPnP/NAT-PMP

- Added: UPnP/NAT-PMP STUN configuration options #10587

Upgrade

- Changed: `pfSense-upgrade` should reinstall all packages on new version upgrades #12235

User Manager / Privileges

- Added: Copy button for group entries in the User Manager #12226

Web Interface

- Changed: Update font formats to WOFF2 #11507
- Fixed: Notifications page cannot be saved without configuring or disabling SMTP #12107
- Changed: Convert help shortcut links to server-side redirects #12314

Wireless

- Fixed: `wpa_supplicant` uses 100% of a CPU core at boot #11453
- Fixed: Interfaces page does not show Wireless EAP client options #12239

XMLRPC

- Added: XMLRPC synchronization for DHCP relay settings #11957
- Changed: XMLRPC client improvements #12051
- Fixed: Changes to an existing IPsec configuration are not applied on HA secondary after XMLRPC sync #12075

3.2.2 21.05.1 New Features and Changes

This is a maintenance release including bug fixes for issues affecting pfSense Plus software version 21.05.

Security

This release includes corrections for the following vulnerabilities in pfSense software:

- Additional corrections for pfSense-SA-21_02.captiveportal (XSS in Captive Portal client login page, #11843)

General

pfSense Plus

FreeBSD

- Fixed: 32-bit ARM performance regression #12200

Operating System

- Changed: Native hardware package builds for 32-bit ARM #12201

PHP Interpreter

- Changed: Disable PCRE JIT to work around PHP PCRE crashes on multi-core 32-bit ARM systems #12004

Routing

- Fixed: Static routes may not be in routing table when expected #11986

3.2.3 2.5.2 New Features and Changes

This is a regularly scheduled software release including new features and bug fixes.

Known Issues / Errata

- Dynamic DNS incorrectly encodes NoIP update credentials #12021

Security

This release includes corrections for the following vulnerabilities in pfSense software:

- pfSense-SA-21_02.captiveportal (XSS in Captive Portal client login page, #11843)

General

- Added: WireGuard **experimental** add-on package

pfSense CE

Aliases / Tables

- Added: PHP shell playback script to modify Alias contents #11380

Authentication

- Added: Copy button for Authentication Server entries #11390

Backup / Restore

- Added: Randomize time of scheduled AutoConfigBackup runs #10811
- Fixed: Automated corruption recovery from cached `config.xml` backup files should check multiple backups #11748
- Fixed: AutoConfigBackup schedule custom hour value lost on page load #11946

Captive Portal

- Added: Redirect Captive Portal users to login page after they logout #11264
- Fixed: Captive Portal post-auth redirect is not properly respected #11842
- Fixed: Potential XSS vulnerability in Captive Portal `redirurl` handling #11843

Certificates

- Fixed: Certificate Manager does not report Unbound as using a certificate #11678
- Fixed: PHP error on certificate list due to unreadable private key #11859
- Fixed: Export P12 icon is missing if certificate is not locally renewable #11884

Configuration Upgrade

- Fixed: PHP error in `upgrade_212_to_213()` when upgrading certain IPsec tunnels #11801

Console Menu

- Changed: Allow reroot on ZFS from console and GUI reboot menu entries #11914

DHCP (IPv6)

- Fixed: `dhcp6withoutra_script.sh` does not get executed when advanced options are set #11883

DNS Forwarder

- Fixed: Disable DNSSEC option for dnsmasq #11781
- Fixed: Update dnsmasq to 2.85 to fix CVE-2021-3448 #11866

DNS Resolver

- Fixed: Unbound Python Integration repeatedly mounts `dev` without unmounting #11456
- Fixed: Stale hostname registration data for OpenVPN clients is not deleted from the DNS Resolver configuration at boot #11704
- Changed: Temporarily move back to Unbound 1.12.x due to instability on Unbound 1.13.x #11915

Dashboard

- Fixed: Thermal sensors widget no longer shows values from certain hardware #11787
- Fixed: IPsec Dashboard widget only displays first P2 subnet when using a single traffic selector #11893
- Fixed: Editing widgets on Dashboard causes a PHP Warning #11939

Diagnostics

- Fixed: ARP Table populates hostname values using expired DHCP lease data #11510
- Fixed: Sanitize OpenVPN Client Export certificate password in status output #11767
- Fixed: Sanitize Captive Portal RADIUS MAC secret in status output #11769
- Fixed: MAC address OEM information missing from ARP table #11819
- Fixed: State table content on `diag_dump_states.php` does not sort properly #11852

Dynamic DNS

- Added: New Dynamic DNS Provider: Mythic-Beasts #7842
- Added: New Dynamic DNS Provider: one.com #11293
- Added: New Dynamic DNS Provider: Yandex PDD #11294
- Added: New Dynamic DNS Provider: NIC.RU #11358
- Added: New Dynamic DNS Provider: Gandi LiveDNS IPv6 #11420
- Fixed: Automatic 25-day forced Dynamic DNS update removes wildcard domain #11667
- Fixed: Digital Ocean Dynamic DNS help text is incorrect #11754
- Fixed: NoIP.com Dynamic DNS update failure is not detected properly #11815
- Fixed: Dynamic DNS edit page incorrectly hides username field when switching away from Digital Ocean #11840

Gateways

- Added: Input validation to prevent setting a load balancing gateway group as default #11164

Hardware / Drivers

- Changed: Deprecate old cryptographic accelerator hardware which is not viable on modern systems #11426
- Fixed: Using SHA1 or SHA256 with AES-NI may fail if AES-NI attempts to accelerate hashing #11524

High Availability

- Fixed: Incorrect RADVD log message on HA event #11966

IGMP Proxy

- Fixed: IGMP Proxy restarts unnecessarily after IPv6 gateway events #11904

IPsec

- Added: GUI option to set RADIUS Timeout for EAP-RADIUS #11211
- Added: Option to switch IPsec filtering modes to choose between `enc` and `if_ipsec` filtering #11395
- Changed: Move custom IPsec NAT-T port settings to Advanced Options #11518
- Fixed: strongSwan configuration always contains user EAP/PSK values #11564
- Added: IPsec GUI option to control Child SA `start_action` #11576
- Fixed: Error when adding both IPv4 and IPv6 P2 under an IPv4 or IPv6 only IKEv1 P1 #11651
- Fixed: Cannot disable IPsec P1 when related P2s are in VTI mode and enabled #11792
- Fixed: IPsec VTI interface names are not properly formed for more than 32 interfaces #11794
- Fixed: Applying IPsec settings for more than ~30 tunnels times out PHP #11795
- Fixed: `ipsec_vti()` does not skip disabled VTI entries #11832
- Fixed: IPsec GUI allows creating multiple identical Phase 1 entries when using FQDN for remote gateway #11912
- Fixed: Mobile IPsec advanced RADIUS parameters do not allow numeric values with a decimal point #11967

IPv6 Router Advertisements (RADVD)

- Added: Use virtual link local IP address as RA source address for HA environments #11103
- Added: Shortcut buttons for service control and logs on RADVD configuration #11911
- Fixed: RADVD breaks on SIGHUP #11913

Interfaces

- Fixed: DHCP interfaces are always treated as having a gateway, even if one is not assigned by the upstream DHCP server #5135
- Fixed: Interfaces page displays MAC Address field for interfaces which do not support L2 #11387
- Fixed: CLI interface configuration without IPv6 leaves RA enabled #11609
- Fixed: Incomplete PPPoE custom reset values lead to invalid cron entry #11698
- Fixed: Error when changing MTU if the interface is used for both IPv4 and IPv6 default routes #11855
- Added: VLAN list sorting #11968

L2TP

- Fixed: Unused L2TP VPN files are not removed when the service is disabled #11299
- Added: GUI option to set MTU for L2TP VPN server #11406

NTPD

- Fixed: NTP widget displays incorrect status #11495
- Fixed: NTP authentication input validation rejects valid keys #11850

Notifications

- Fixed: Invalid HTML encoding in modal Notices window #11765

OpenVPN

- Added: Allow the firewall to use DNS servers provided to an OpenVPN client instance #11140
- Fixed: OpenVPN Wizard does not support gateway groups #11141
- Added: Set Explicit Exit Notify to `1` by default for new OpenVPN client instances #11521
- Added: Support for Cisco AVPair `{clientipv6}` template in firewall rules returns by RADIUS #11596
- Changed: Set `explicit-exit-notify` option by default for new OpenVPN server instances #11684
- Fixed: OpenVPN does not clean up parsed `Cisco-AVPair` rules on non-graceful disconnect #11699
- Fixed: OpenVPN does not kill IPv6 client states on disconnect #11700
- Fixed: OpenVPN client starts when CARP VIP is in BACKUP status when bound to Virtual IP aliased to CARP VIP #11793
- Fixed: Certificate validation with OCSP always fails in `openvpn.tls-verify.php` #11830
- Changed: Update OpenVPN to 2.5.2 #11844
- Fixed: OpenVPN client startup error if IPv6 Tunnel Network is defined in TAP mode #11869

Operating System

- Added: Kernel modules for alternate congestion control algorithms #7092
- Added: Kernel module for RTL8153 driver #11125
- Added: Xen console support #11402
- Fixed: Unquoted variable in `dot.tcshrc` can cause proxy password to be printed #11867

Routing

- Fixed: IPv4 link-local (`169.254.x.x`) gateway does not function #11806

Rules / NAT

- Added: Support for IPv6 firewall entries with dynamic delegated prefix and static host address #6626
- Fixed: Disabling all interfaces associated with a floating rule causes the firewall to generate an incorrect pf rule #11688
- Fixed: Input validation prevents creating 1:1 NAT rules on IPsec #11751
- Fixed: Invalid combinations of TCP flag matching options cause `pfctl` parser error #11762
- Fixed: Port forward rules only function through the default gateway interface, `reply-to` does not work for Multi-WAN (CE Only) #11805
- Fixed: Error loading rules in certain cases where an interface is temporarily without an address #11861
- Fixed: NAT 1:1 fail to validate aliases #11923

Traffic Shaper (ALTQ)

- Fixed: Harmless error when enabling traffic shaper #11229
- Fixed: Segmentation fault when loading ALTQ traffic shaping rules using FAIRQ #11550

Traffic Shaper (Limiters)

- Fixed: Unused Limiter entries with schedules create unnecessary cron jobs #11636
- Fixed: Error when setting queue limit on CODELQ limiter #11725

Upgrade

- Fixed: Language presented to user during upgrade is misleading #11897

Web Interface

- Added: Replace HTTP links with HTTPS in the GUI #11228
- Fixed: Ambiguous text in help and input validation error for system domain name #11658
- Fixed: PHP error if `PHP_error.log` file is too large #11685
- Fixed: RAM Disk Settings shows Kernel Memory at `0` Kb and does not allow the user to create RAM disks #11702
- Fixed: HTTP Referer error message text is incorrect #11873
- Fixed: Missing `/0` subnet when cloning repeatable CIDR mask controls #11880
- Fixed: Update NGINX to address CVE-2021-23017 #12061

WireGuard

- Fixed: Ignore WireGuard configurations under `<installedpackages></installedpackages>` #11808

Wireless

- Added: GUI options for WPA Enterprise with identity/password #2400
- Fixed: `wpa_supplicant` uses 100% of a CPU core at boot #11453

XMLRPC

- Fixed: XMLRPC synchronization restarts all OpenVPN instances on the secondary node when making any change on the primary node #11082
- Fixed: XMLRPC Client does not honor its default timeout value #11718

3.2.4 21.05 New Features and Changes

This is a regularly scheduled software release of pfSense Plus software including new features, additional hardware support, and bug fixes.

Security

This release includes corrections for the following vulnerabilities in pfSense software:

- pfSense-SA-21_02.captiveportal (XSS in Captive Portal client login page, #11843)

General

- Added: WireGuard **experimental** add-on package
- Added: OpenVPN client import add-on package
- Fixed: `ix(4)` driver fails to attach if a broken or unsupported SFP module (e.g. incompatible media type) is present at boot time [NG 1586]
- Fixed: IP Address ranges do not work in aliases on 32-bit ARM [NG 5445]

pfSense Plus

Aliases / Tables

- Added: PHP shell playback script to modify Alias contents #11380

Authentication

- Added: Copy button for Authentication Server entries #11390

Backup / Restore

- Added: Randomize time of scheduled AutoConfigBackup runs #10811
- Fixed: Automated corruption recovery from cached `config.xml` backup files should check multiple backups #11748

Captive Portal

- Added: Redirect Captive Portal users to login page after they logout #11264
- Fixed: Captive Portal post-auth redirect is not properly respected #11842
- Fixed: Potential XSS vulnerability in Captive Portal `redirurl` handling #11843

Certificates

- Fixed: Certificate Manager does not report Unbound as using a certificate #11678
- Fixed: PHP error on certificate list due to unreadable private key #11859
- Fixed: Export P12 icon is missing if certificate is not locally renewable #11884

Configuration Upgrade

- Fixed: PHP error in `upgrade_212_to_213()` when upgrading certain IPsec tunnels #11801

Console Menu

- Changed: Allow reroot on ZFS from console and GUI reboot menu entries #11914

DHCP (IPv6)

- Fixed: `dhcp6withoutra_script.sh` does not get executed when advanced options are set #11883

DNS Forwarder

- Fixed: Disable DNSSEC option for dnsmasq #11781
- Fixed: Update dnsmasq to 2.85 to fix CVE-2021-3448 #11866

DNS Resolver

- Fixed: Unbound Python Integration repeatedly mounts `dev` without unmounting #11456
- Fixed: Stale hostname registration data for OpenVPN clients is not deleted from the DNS Resolver configuration at boot #11704
- Changed: Temporarily move back to Unbound 1.12.x due to instability on Unbound 1.13.x #11915

Dashboard

- Fixed: Thermal sensors widget no longer shows values from certain hardware #11787
- Fixed: IPsec Dashboard widget only displays first P2 subnet when using a single traffic selector #11893
- Fixed: Editing widgets on Dashboard causes a PHP Warning #11939

Diagnostics

- Fixed: ARP Table populates hostname values using expired DHCP lease data #11510
- Fixed: Sanitize OpenVPN Client Export certificate password in status output #11767
- Fixed: Sanitize Captive Portal RADIUS MAC secret in status output #11769
- Fixed: MAC address OEM information missing from ARP table #11819

Dynamic DNS

- Added: New Dynamic DNS Provider: Mythic-Beasts #7842
- Added: New Dynamic DNS Provider: one.com #11293
- Added: New Dynamic DNS Provider: Yandex PDD #11294
- Added: New Dynamic DNS Provider: NIC.RU #11358
- Added: New Dynamic DNS Provider: Gandi LiveDNS IPv6 #11420
- Fixed: Automatic 25-day forced Dynamic DNS update removes wildcard domain #11667
- Fixed: Digital Ocean Dynamic DNS help text is incorrect #11754
- Fixed: NoIP.com Dynamic DNS update failure is not detected properly #11815
- Fixed: Dynamic DNS edit page incorrectly hides username field when switching away from Digital Ocean #11840

Gateways

- Added: Input validation to prevent setting a load balancing gateway group as default #11164

Hardware / Drivers

- Changed: Deprecate old cryptographic accelerator hardware which is not viable on modern systems #11426
- Fixed: Using SHA1 or SHA256 with AES-NI may fail if AES-NI attempts to accelerate hashing #11524

IGMP Proxy

- Fixed: IGMP Proxy restarts unnecessarily after IPv6 gateway events #11904

IPsec

- Added: GUI option to set RADIUS Timeout for EAP-RADIUS #11211
- Added: Option to switch IPsec filtering modes to choose between `enc` and `if_ipsec` filtering #11395
- Changed: Move custom IPsec NAT-T port settings to Advanced Options #11518
- Fixed: strongSwan configuration always contains user EAP/PSK values #11564
- Added: IPsec GUI option to control Child SA `start_action` #11576
- Fixed: Error when adding both IPv4 and IPv6 P2 under an IPv4 or IPv6 only IKEv1 P1 #11651
- Fixed: Cannot disable IPsec P1 when related P2s are in VTI mode and enabled #11792
- Fixed: IPsec VTI interface names are not properly formed for more than 32 interfaces #11794
- Fixed: Applying IPsec settings for more than ~30 tunnels times out PHP #11795
- Fixed: `ipsec_vti()` does not skip disabled VTI entries #11832
- Fixed: IPsec GUI allows creating multiple identical Phase 1 entries when using FQDN for remote gateway #11912

IPv6 Router Advertisements (RADVD)

- Added: Use virtual link local IP address as RA source address for HA environments #11103
- Added: Shortcut buttons for service control and logs on RADVD configuration #11911
- Fixed: RADVD breaks on SIGHUP #11913

Interfaces

- Fixed: DHCP interfaces are always treated as having a gateway, even if one is not assigned by the upstream DHCP server #5135
- Fixed: Interfaces page displays MAC Address field for interfaces which do not support L2 #11387
- Fixed: CLI interface configuration without IPv6 leaves RA enabled #11609
- Fixed: Incomplete PPPoE custom reset values lead to invalid cron entry #11698
- Fixed: Error when changing MTU if the interface is used for both IPv4 and IPv6 default routes #11855

L2TP

- Fixed: Unused L2TP VPN files are not removed when the service is disabled #11299
- Added: GUI option to set MTU for L2TP VPN server #11406

NTPD

- Fixed: NTP widget displays incorrect status #11495
- Fixed: NTP authentication input validation rejects valid keys #11850

Notifications

- Fixed: Invalid HTML encoding in modal Notices window #11765

OpenVPN

- Added: Allow the firewall to use DNS servers provided to an OpenVPN client instance #11140
- Fixed: OpenVPN Wizard does not support gateway groups #11141
- Added: Set Explicit Exit Notify to `1` by default for new OpenVPN client instances #11521
- Added: Support for Cisco AVPair `{clientipv6}` template in firewall rules returns by RADIUS #11596
- Fixed: OpenVPN does not clean up parsed `Cisco-AVPair` rules on non-graceful disconnect #11699
- Fixed: OpenVPN does not kill IPv6 client states on disconnect #11700
- Fixed: OpenVPN client starts when CARP VIP is in BACKUP status when bound to Virtual IP aliased to CARP VIP #11793
- Fixed: Certificate validation with OCSP always fails in `openvpn.tls-verify.php` #11830
- Changed: Update OpenVPN to 2.5.2 #11844
- Fixed: OpenVPN client startup error if IPv6 Tunnel Network is defined in TAP mode #11869

Operating System

- Added: Kernel modules for alternate congestion control algorithms #7092
- Added: Kernel module for RTL8153 driver #11125
- Added: Xen console support #11402
- Fixed: Unquoted variable in `dot.tcshrc` can cause proxy password to be printed #11867

Routing

- Fixed: Static route targets may still reachable via default route when the gateway they should route through is down #11296
- Fixed: IPv4 link-local (`169.254.x.x`) gateway does not function #11806

Rules / NAT

- Added: Support for IPv6 firewall entries with dynamic delegated prefix and static host address #6626
- Fixed: Disabling all interfaces associated with a floating rule causes the firewall to generate an incorrect pf rule #11688
- Fixed: Input validation prevents creating 1:1 NAT rules on IPsec #11751
- Fixed: Invalid combinations of TCP flag matching options cause `pfctl` parser error #11762
- Fixed: Error loading rules in certain cases where an interface is temporarily without an address #11861

Traffic Shaper (ALTQ)

- Fixed: Harmless error when enabling traffic shaper #11229
- Fixed: Segmentation fault when loading ALTQ traffic shaping rules using FAIRQ #11550

Traffic Shaper (Limiters)

- Fixed: Unused Limiter entries with schedules create unnecessary cron jobs #11636
- Fixed: Error when setting queue limit on CODELQ limiter #11725

Upgrade

- Fixed: Language presented to user during upgrade is misleading #11897

Web Interface

- Added: Replace HTTP links with HTTPS in the GUI #11228
- Fixed: Ambiguous text in help and input validation error for system domain name #11658
- Fixed: PHP error if `PHP_error.log` file is too large #11685
- Fixed: RAM Disk Settings shows Kernel Memory at `0` Kb and does not allow the user to create RAM disks #11702
- Fixed: HTTP Referer error message text is incorrect #11873
- Fixed: Missing `/0` subnet when cloning repeatable CIDR mask controls #11880

WireGuard

- Fixed: Ignore WireGuard configurations under `<installedpackages></installedpackages>` #11808

Wireless

- Added: GUI options for WPA Enterprise with identity/password #2400

XMLRPC

- Fixed: XMLRPC synchronization restarts all OpenVPN instances on the secondary node when making any change on the primary node #11082
- Fixed: XMLRPC Client does not honor its default timeout value #11718

3.2.5 21.02.2/2.5.1 New Features and Changes

pfSense Plus software version 21.02.2 and pfSense CE software version 2.5.1 are maintenance releases to address recently identified issues.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

Tip: To remove WireGuard tunnels, navigate to **VPN > WireGuard** and click the delete button for each tunnel. When the page displays **No WireGuard tunnels have been configured.**, the upgrade can proceed.

Note: This pfSense Plus software version contains all of the items noted below for pfSense CE as well.

Tip: For those who have not yet updated to 2.4.5-p1, consult the *previous release notes* and blog posts for those releases to read all important information and warnings before proceeding.

Known Issues / Errata

- There is an issue in this release with port forwarding on pfSense CE software installations with multiple WANs, see #11805 for details.
- There is an issue with AES-NI hash acceleration for SHA1 and SHA-256. If the AES-NI driver detects a system capable of accelerating SHA1 or SHA-256 and the firewall attempts to utilize one of those hashes, the affected operation may fail. This affects IPsec and OpenVPN, among other uses. pfSense Plus users can change to QAT acceleration on supported hardware instead. In cases where QAT is unavailable, change to AES-GCM, change to a different unaccelerated hash (e.g. SHA-512), or disable AES-NI. See #11524 for details.
- There is a similar issue which affects SafeXcel SHA1 and SHA2 hash acceleration on SG-1100 and SG-2100. On that hardware, change to an AEAD cipher such as AES-GCM or switch to an unaccelerated hash. This issue is being tracked internally on NG #6005
- The FRR package on pfSense Plus 21.02 and pfSense CE 2.5.0 and later no longer exchanges routes with BGP peers by default without being explicitly allowed to do so. This is more secure behavior but requires a manual change. To replicate the previous behavior, use **ONE** of the following workarounds:
  - Navigate to **Services > FRR BGP** on the **Advanced** tab and check *Disable eBGP Require Policy*, then **Save**.
  - Instead of disabling the policy check, create route maps which match and allow expected incoming and outgoing routes explicitly. This is the most secure method. See *Peer Filtering* and *BGP Example Configuration* for more information.
  - Manually create a route map to permit all routes (Name: `allow-all`, Action: *Permit*, Sequence: `100`), then set that route map on BGP neighbors for inbound and outbound peer filtering. This can be used as a placeholder for later migration to more secure route map filtering.

Security

This release includes corrections for the following vulnerabilities in pfSense software:

- pfSense-SA-21_01.webgui (XSS in Wake on LAN, #11616)

General

pfSense Plus

Certificates

- Fixed: CA and certificate validity end dates after 2038 are not handled properly on 32-bit ARM #11504

Interfaces

- Added: Interface Status page information for switch uplinks may be replaced by switch port data when media state monitoring is set #10804

Rules / NAT

- Fixed: State matching problem with reponses to packets arriving on non-default WANs #11436

Upgrade

- Fixed: LEDs do not indicate available upgrade status #11689

pfSense CE

Aliases / Tables

- Fixed: Alias name change is not reflected in firewall rules #11568

Authentication

- Fixed: Unreachable LDAP server for SSH auth causes boot process to stop at at 'Synchronizing user settings' and no user can login over SSH #11644

Certificates

- Fixed: Invalid certificate data can cause a PHP error #11489
- Fixed: Renewing a self-signed CA or certificate does not update the serial number #11514
- Fixed: Unable to renew a certificate without a SAN #11652
- Fixed: Certificates with escaped x509 characters display the escaped version when renewing #11654
- Fixed: Creating a certificate while creating a user does not fully configure the certificate properly #11705
- Fixed: Renewing a certificate without a `type` value assumes a server certificate #11706

DNS Resolver

- Fixed: DNS Resolver does not add a `local-zone` type for `ip6.arpa` domain override #11403
- Fixed: DNS Resolver does not bind to an interface when it recovers from a down state #11547

Dashboard

- Fixed: CPU details are incorrect in the System Information widget after resetting log files #11428
- Fixed: Disabling 'State Table Size' in the System Information widget prevents other data from being displayed #11443

Gateway Monitoring

- Fixed: Automatic default gateway mode does not select expected entries #11729

Gateways

- Fixed: Gateways with "Use non-local gateway" set are not added to routing table #11433

IPsec

- Fixed: IPsec status incorrect for entries using expanded IKE connection numbers #11435
- Fixed: Distinguished Name (FQDN) IPsec peer identifier type is not formatted properly in `swanctl.conf` secrets #11442
- Fixed: Mobile IPsec DNS server input validation does not reject unsupported IPv4-mapped IPv6 addresses #11446
- Fixed: Broken help link on IPsec Advanced Settings tab #11474
- Fixed: Connect and disconnect buttons on the IPsec status page do not work for all tunnels #11486
- Fixed: IPsec tunnels using expanded IKE connection numbers do not have proper child SA names in `swanctl.conf` #11487
- Fixed: IPsec tunnel definitions have `pools =` entry in `swanctl.conf` with no value #11488
- Fixed: Mobile IPsec broken when using strict certificate revocation list checking #11526
- Fixed: IPsec VTI tunnel between IPv6 peers may not configure correctly #11537
- Fixed: IPsec peer ID of "Any" does not generate a proper remote definition or related secrets #11555
- Fixed: IPsec tunnel does not function when configured on a 6RD interface #11643

IPv6 Router Advertisements (RADVD)

- Fixed: IPv6 RA RDNSS lifetime is too short, not compliant with RFC 8106 #11105

Installer

- Fixed: Installer does not add required module to loader.conf when using ZFS #11483

Interfaces

- Fixed: IPv4 MSS value is incorrectly applied to IPv6 packets #11409
- Fixed: Gateway value for DHCP6 interfaces missing after RA events triggered script without gateway information #11454
- Fixed: Delayed packet transmission in cxgbe driver can lead to latency and reduced performance #11602
- Fixed: DHCP6 interfaces are reconfigured multiple times at boot when more than one interface is set to Track #11633

Logging

- Fixed: Entries from rotated log files may be displayed out of order when log display includes contents from multiple files #11639

Notifications

- Fixed: Telegram and Pushover notification API calls do not respect proxy configuration #11476

OpenVPN

- Fixed: OpenVPN authentication and certificate validation fail due to size of data passed through `fcgicli` #4521
- Added: Display negotiated data encryption algorithm in OpenVPN connection status #7077
- Fixed: OpenVPN does not start with several authentication sources selected #11104
- Fixed: OpenVPN client configuration page displays Shared Key option when set for SSL/TLS #11382
- Fixed: Incorrect order of `route-nopull` option in OpenVPN client-specific override configuration #11448
- Fixed: OpenVPN using the wrong OpenSSL command to list digest algorithms #11500
- Fixed: Selected Data Encryption Algorithms list items reset when an input validation error occurs #11554
- Fixed: OpenVPN does not start with a long list of Data Encryption Algorithms #11559
- Fixed: ACLs generated from RADIUS reply attributes do not parse `{clientip}` macro #11561
- Fixed: ACLs generated from RADIUS reply attributes have incorrect syntax #11569
- Fixed: OpenVPN binds to all interfaces when configured on a 6RD interface #11674

Operating System

- Fixed: Unexpected Operator error on console at boot with ZFS and RAM Disks #11617
- Changed: Upgrade OpenSSL to 1.1.1k #11755

Routing

- Fixed: Disabled static route entries trigger 'route delete' error at boot #3709
- Fixed: Route tables with many entries can lead to PHP errors and timeouts when looking up routes #11475
- Fixed: Error when removing automatic DNS server route #11578
- Fixed: IPv6 routes with a prefix length of 128 result in an invalid route table entry #11594
- Fixed: Error when deleting IPv6 link-local routes #11713

Rules / NAT

- Fixed: Saved state timeout values not loaded into GUI fields on system_advanced_firewall.php #11565
- Fixed: Firewall rule schedule cannot be changed #11747

Upgrade

- Fixed: pfSense Proxy Authentication not working #11383

Wake on LAN

- Fixed: Potential stored XSS vulnerability in services_wol.php #11616

Web Interface

- Fixed: Requests to `ews.netgate.com` do not honor proxy configuration #11464

XMLRPC

- Fixed: XMLRPC error with Captive Portal and CARP failover when GUI is on non-standard port #11425
- Fixed: Incorrect DHCP failover IP address configured on peer after XMLRPC sync #11519
- Fixed: PHP error in logs from XMLRPC if no sections are selected to sync #11638

3.2.6 21.02/21.02-p1/2.5.0 New Features and Changes

pfSense® Plus software version 21.02 and pfSense Community Edition (CE) software version 2.5.0 include a major OS version upgrade, a kernel WireGuard implementation, OpenSSL upgrades, VPN and related security improvements, plus numerous other bug fixes and new features.

Warning: The original plan was to include a RESTCONF API in pfSense® Plus software version 21.02 and pfSense software version 2.5.0, which for security reasons would have required hardware AES-NI or equivalent cryptographic accelerator support. Plans have since changed, and these versions do not contain the planned RESTCONF API, thus **pfSense® Plus software version 21.02 and pfSense Community Edition (CE) software version 2.5.0 DO NOT require AES-NI**.

Tip: For those who have not yet updated to 2.4.5-p1, consult the *previous release notes* and blog posts for those releases to read all important information and warnings before proceeding.

pfSense Plus

Version 21.02 is the first release of pfSense Plus software, formerly known as Factory Edition. For more details about the distinctions between pfSense Plus and pfSense CE, read the pfSense Plus Announcement. Customers running the Factory Edition of pfSense software version 2.4.5-p1 and older can upgrade in-place automatically to pfSense Plus software version 21.02 as with any other previous upgrade.

In this version, the changes in pfSense Plus software and pfSense CE software are roughly the same, with a few notable exceptions which are only available in pfSense Plus software:

- Support for Intel® QuickAssist Technology, also known as QAT.
  - QAT accelerates cryptographic and hashing operations on supported hardware, and can be used to accelerate IPsec, OpenVPN, and other OpenCrypto Framework-aware software.
  - Supported hardware includes many Intel-based systems sold by Netgate (e.g. XG-7100, SG-5100) and add-on cards.
  - From the FreeBSD man page:
    - The `qat` driver supports the QAT devices integrated with Atom C2000 and C3000 and Xeon C620 and D-1500 chipsets, and the Intel QAT Adapter 8950.
    - It can accelerate AES in CBC, CTR, XTS (except for the C2000) and GCM modes, and can perform authenticated encryption combining the CBC, CTR and XTS modes with SHA1-HMAC and SHA2-HMAC. The qat driver can also compute SHA1 and SHA2 digests.
- Improved SafeXcel cryptographic accelerator support for SG-2100 and SG-1100 which can improve IPsec performance.
  - From the FreeBSD man page:
    - The driver can accelerate the following AES modes: AES-CBC, AES-CTR, AES-XTS, AES-GCM, AES-CCM
    - The driver also implements SHA1 and SHA2 transforms, and can combine AES-CBC and AES-CTR with SHA1-HMAC and SHA2-HMAC for encrypt-then-authenticate operations.
- Updated *IPsec profile export*
  - Exports Apple profiles compatible with current iOS and OS X versions

  - New export function for Windows clients to configure tunnels using PowerShell

Version 21.02-p1

pfSense Plus software version 21.02-p1 is a special patch release to address a kernel problem affecting the SG-3100 which caused system instability (#11444). No additional fixes are present in the 21.02-p1 release.

See the detailed bug analysis blog post for more details.

Operating System / Architecture changes

- Base OS upgraded to FreeBSD **12.2-STABLE**
- OpenSSL upgraded to **1.1.1i-freebsd**
- PHP upgraded to **7.4** #9365 #10659
- Python upgraded to **3.7** #9360

Known Issues / Errata

- Deprecated the built-in relayd Load Balancer #9386
  - `relayd` does not function with OpenSSL 1.1.x
  - The `relayd` FreeBSD port has been changed to require libressl – There is no apparent sign of work to make it compatible with OpenSSL 1.1.x
  - The HAProxy package may be used in its place; It is a much more robust and more feature-complete load balancer and reverse proxy
  - For more information on implementing HAProxy, see *HAProxy package* and the Hangout
- There is an issue in this release with port forwarding on pfSense Plus software installations with multiple WANs, which has been resolved in the 21.02.2 patch release, see #11436 for details.
- There is an issue with AES-NI hash acceleration for SHA1 and SHA-256. If the AES-NI driver detects a system capable of accelerating SHA1 or SHA-256 and the firewall attempts to utilize one of those hashes, the affected operation may fail. This affects IPsec and OpenVPN, among other uses. pfSense Plus users can change to QAT acceleration on supported hardware instead. In cases where QAT is unavailable, change to AES-GCM, change to a different unaccelerated hash (e.g. SHA-512), or disable AES-NI. See #11524 for details.
- There is a similar issue which affects SafeXcel SHA1 and SHA2 hash acceleration on SG-1100 and SG-2100. On that hardware, change to an AEAD cipher such as AES-GCM or switch to an unaccelerated hash. This issue is being tracked internally on NG #6005
- The FRR package on pfSense Plus 21.02 and pfSense CE 2.5.0 and later no longer exchanges routes with BGP peers by default without being explicitly allowed to do so. This is more secure behavior but requires a manual change. To replicate the previous behavior, use **ONE** of the following workarounds:
  - Navigate to **Services > FRR BGP** on the **Advanced** tab and check *Disable eBGP Require Policy*, then **Save**.
  - Instead of disabling the policy check, create route maps which match and allow expected incoming and outgoing routes explicitly. This is the most secure method. See *Peer Filtering* and *BGP Example Configuration* for more information.

  - Manually create a route map to permit all routes (Name: `allow-all`, Action: *Permit*, Sequence: `100`), then set that route map on BGP neighbors for inbound and outbound peer filtering. This can be used as a placeholder for later migration to more secure route map filtering.

Warning: See the FreeBSD 12.0 Release Notes for information on deprecated hardware drivers that may impact firewalls upgrading to pfSense software version 2.5.0. Some of these were renamed or folded into other drivers, others have been removed, and more are slated for removal in FreeBSD 13 in the future.

Aliases/Tables

- Fixed aliases to allow IPv6 prefix entries which end in IPv4 addresses (e.g. `x:x:x:x:x:x:d.d.d.d` from RFC 4291 section 2.2.2) #10694
- Fixed a PHP error processing aliases when the configuration contains no aliases section #9936
- Fixed URL-based Alias only storing last-most entry in the configuration #9074
- Fixed an issue with PF tables remaining active after they had been deleted #9790
- Added Internationalized domain names support for aliases #7255
- Added the ability to copy an existing alias when creating a new entry #6908
- Fixed handling of URL-based aliases containing multiple URLs #11256

Authentication

- Added RADIUS authentication for SSH users #10545
- Added LDAP authentication for SSH users #8698
- Added option to control behavior of unauthenticated LDAP binds #9909
- Converted LDAP TLS setup from environment variables to LDAP_OPT_X_TLS_* options #9417
- Set RADIUS NAS Identifier to include `webConfigurator` and the firewall hostname when logging in the GUI #9209
- Added LDAP extended query for groups in RFC2307 containers #9527
- Fixed errors when using RADIUS for GUI authentication while the WAN is down #11109

Backup/Restore

- Changed `crypt_data()` to use stronger key derivation #9421
- Updated `crypt_data()` syntax for OpenSSL 1.1.x #9420 #10178
- Disabled AutoConfigBackup manual backups when AutoConfigBackup is disabled #9785
- Improved error handling when attempting to restore encrypted and otherwise invalid configurations which result in errors (e.g. wrong encryption passphrase, malformed XML) #10179
- Added option to include the DHCP v4/v6 leases database in config.xml backups #10910
- Added option to include the Captive Portal database in config.xml backups #10868
- Added option to include the Captive Portal used MACs database in config.xml backups #10856
- Added option to prevent all extra data from being added to config.xml backups #10914

- Added password confirmation when encrypting a config.xml backup #10301
- Added support for GPT partitioned drives to the External Configuration Locator #9097
- Added support for Limiters to the Traffic Shaper backup and restore area option #4763
- Added option to backup Dynamic DNS area #3559
- Fixed restoration of active voucher data from backup #3128

Captive Portal

- Improved XMLRPC sync of Captive Portal database information #97
- Changed Captive Portal vouchers to use `phpseclib` so it can generate keys natively in PHP, and to work around OpenSSL deprecating key sizes needed for vouchers #9443
- Added `trim()` to the submitted username, so that spaces before/after in input do not cause authentication errors #9274
- Optimized Captive Portal authentication attempts when using multiple authentication servers #9255
- Fixed Captive Portal session timeout values for RADIUS users who do not have a timeout returned from the server #9208
- Changed Captive Portal so that users no longer get disconnected when changes are made to Captive Portal settings #8616
- Added an option so that Captive Portals may choose to remove or retain logins across reboot #5644
- Fixed deletion of related files when removing a Captive Portal zone #10891
- Fixed XMLRPC sync of Captive Portal used MACs database #10857
- Added validation of Captive Portal zone names to prevent using reserved words #10798
- Added support for IDN hostnames to Captive Portal Allowed Hostnames tab #10747
- Improved Captive Portal Allowed Hostnames so it supports multiple DNS records in responses #10724
- Fixed retention of automatic pass-through MAC entries when using Captive Portal Vouchers #9933
- Fixed Captive Portal Bandwidth per-user bandwidth limit values being applied when disabled #9437 #9311
- Changed handling of voucher logins with Concurrent Login option so that new logins are prevented rather than removing old sessions #9432 #2146
- Changed XMLRPC behavior to not remove zones from secondary node when disabling Captive Portal #9303
- Fixed XMLRPC sync failing to propagate voucher roll option changes to the secondary node #8809
- Fixed XMLRPC sync failing to create Captive Portal voucher files on secondary node #8807
- Fixed Captive Portal + Bridge interface validation #6528
- Added support for masking of Captive Portal pass-thru MACs #2424
- Added support for pre-filling voucher codes via URL parameters, so they can be used via QR code #1984

Certificates

- Fixed OCSP stapling detection for OpenSSL 1.1.x #9408
- Fixed GUI detection of revoked status for certificates issued and revoked by an intermediate CA #9924
- Removed PKCS#12 export links for entries which cannot be exported in that format (e.g. no private key) #10284
- Added an option to globally trust local CA manager entries #4068
- Added support for randomized certificate serial numbers when creating or signing certificates with local internal CAs #9883
- Added validation for CA/CRL serial numbers #9883 #9869
- Added support for importing ECDSA keys in certificates and when completing signing requests #9745
- Added support for creating and signing certificates using ECDSA keys #9843 #10658
- Added detailed certificate information block to the CA list, using code shared with the Certificate list #9856
- Added Certificate Lifetime to certificate information block #7332
- Added CA validity checks when attempting to pre-fill certificate fields from a CA #3956
- Added a daily certificate expiration check and notice, with settings to control its behavior and notifications (Default: 27 days) #7332
- Added functionality to import certificates without private keys (e.g. PKCS#11) #9834
- Added functionality to upload a PKCS#12 file to import a certificate #8645
- Added CA/Certificate renewal functionality #9842
  - This allows a CA or certificate to be renewed using its current settings (or a more secure profile), replacing the entry with a fresh one, and optionally retaining the existing key.
- **Added an "Edit" screen for Certificate entries**
  - This view allows editing the Certificate **Descriptive name** field #7861
  - This view also adds a (not stored) password field and buttons for exporting encrypted private keys and PKCS#12 archives #1192
- **Improved default GUI certificate strength and handling of weak values #9825**
  - Reduced the default GUI web server certificate lifetime to 398 days to prevent errors on Apple platforms #9825
  - Added notes on CA/Cert pages about using potentially insecure parameter choices
  - Added visible warnings on CA/Cert pages if parameters are known to be insecure or not recommended
- **Revamped CRL management to be easier to use and more capable**
  - Added the ability to revoke certificates by serial number #9869
  - Added the ability to revoke multiple entries at a time #3258
  - Decluttered the main CRL list screen
  - Moved to a single CRL create control to the bottom under the list rather than multiple buttons
- **Optimized CA/Cert/CRL code in various ways, including:**
  - Actions are now performed by `refid` rather than array index, which is more accurate and not as prone to being affected by parallel changes
  - Improved configuration change descriptions as shown in the GUI and configuration history/backups

  - Miscellaneous style and code re-use improvements
  - Changed CA/Cert date calculations to use a more accurate method, which ensures accuracy on ARM past the 2038 date barrier #9899

Configuration Backend

- Changed error handling on boot error 'XML configuration file not found' so the user is given an opportunity to fix the problem manually #10556

Configuration Upgrade

- Retired m0n0wall configuration upgrade support #10997

Console Menu

- Fixed `rc.initial` execution of `rc.local.running` #10978
- Fixed `rc.initial` handling of `-c` commands with arguments #10603
- Fixed console menu display of subnet masks for DHCP interfaces #10740

Dashboard

- Added PPP uptime to the Dashboard Interfaces Widget #9426
- Improved long description truncation behavior in the services status widget #10795
- Fixed Dashboard traffic graph widget display of bandwidth units (b/s vs. B/s) #9072
- Added adaptive state timeout indication to the state table usage meter #7016
- Fixed Thermal Sensors dashboard widget showing invalid sensors #10963
- Added default route indicator to Gateways widget #11057
- Added hardware interface name as a tooltip on Interfaces widget entries #11041

DHCP (IPv4)

- Fixed handling of spaces in DHCP lease hostnames by `dhcpleases` #9758
- Fixed DHCP leases hostname parsing problems which prevented some hostnames from being displayed in the GUI #3500
- Added OMAPI settings to the DHCP Server #7304
- Increased number of NTP servers sent via DHCP to 3 #9661
- Added an option to prevent known DHCP clients from obtaining addresses on any interface (e.g. known clients may only obtain an address from the interface where the entry is defined) #1605
- Added count of static mappings to list when editing DHCP settings for an interface #9282
- Fixed handling of client identifiers on static mappings containing double quotes #10295
- Added ARM32/64 network booting support to the DHCP Server #10374
- Increased the number of NTP servers for DHCP Static Mappings #10333

- Fix DHCP Dynamic DNS handling of per-host zone and key options from static mappings #10224
- Added per-host custom BOOTP/DHCP Options to static mappings #8990
- Added a button to clear all DHCP leases #7406
- Fixed ARPA zone declaration formatting in DHCP server configuration file #11224

DHCP (IPv6)

- Added options to disable pushing IPv6 DNS servers to clients via DHCP6 #9302
- Fixed DHCPv6 domain search list #10200
- Fixed validation to allow omission of DHCPv6 range for use with stateless DHCP #9596
- Fixed issues creating IPv6 Static Mappings #7443
- Fixed DHCPv6 merging an IPv6 prefix with the input submitted in DNS servers field when using Track Interface #7384
- Fixed prefix delegation not being requested if no interfaces were set to track6 #11005
- Fixed DHCPv6 Dynamic DNS domain key name validation #10844
- Fixed line formatting issues in the DHCPv6 configuration file #10675
- Fixed prefix not being included in the DNS entry registered by DHCPv6 #8156
- Fixed DHCPv6 static mapping changes requiring a restart of the DNS resolver to activate #10882
- Fixed issues running DHCPv6 on certain types of tracked interfaces (e.g. bridges, VLANs) #3965
- Fixed issues with WAN not renewing IPv6 address after an upstream failure #10966

DHCP Relay

- Fixed DHCP Relay validation to allow OpenVPN TAP interfaces #10711
- Fixed inconsistent validation behavior for DHCP relay and bridges #7778

Diagnostics

- Added Reroot and Reboot with Filesystem Check options to GUI Reboot page #9771
- Added option to control wait time between ICMP echo request (ping) packets `diag_ping.php` #9862
- Improved data sanitization in status.php #10946 #10944 Sanitize MaxMind GeoIP key #10797 #10569 #10794
- Added config history list to status.php #10696
- Added DNS Resolver configuration to status.php #10635
- Added L2TP VPN configuration to status.php #10583
- Changed pftop page to hide filtering controls for views which do not support filtering #10625
- Added support for IDN hostnames to DNS Lookup, Ping, and Traceroute #10538
- Fixed diag_dns.php link to Ping passing incorrect parameters #10537
- Added a button to clear the NDP cache #10975
- Added a button to clear the ARP cache #4038

- Fixed hostname being ignored when DNS Lookup calculates response time #11018
- Fixed **Kill States** button on `diag_dump_states.php` when used with CIDR-masked subnets #9270

DNS Forwarder

- Updated `dnsmasq` to 2.84 #11278

DNS Resolver

- Added IPv6 OpenVPN client addresses resolution to the DNS Resolver #8624
- Added DNS64 options to the DNS Resolver #10274
- Added support for multiple IP addresses in a DNS Resolver Host Override entry #10896
- Fixed DNS Resolver restart commands to work around potential environment issues #10781
- Fixed saving DNS Resolver ACL entries when using a non-English translation #10742
- Added support for IDN symbols in DNS Resolver ACL entries #10730
- Added Aggressive NSEC option to the DNS Resolver #10449
- Fixed DNS Resolver unintentionally retaining DHCP registration entries after disabling that feature #8981
- Fixed DNS Resolver restarting on every OpenVPN client connection when registering clients in DNS #11129
- Fixed issues with the DNS Resolver not starting when bound to disabled interfaces or interfaces without carrier #11087
- Fixed DNS Resolver custom TLS listen port being ignored #11051
- Improved formatting and ordering of items in the DNS Resolver access list configuration file #11309

Dynamic DNS

- Fixed Dynamic DNS Dashboard Widget address parsing for entries with split hostname/domain (e.g. Namecheap) #9564
- Added support for new CloudFlare Dynamic DNS API tokens #9639
- Added IPv6 support to No-IP Dynamic DNS #10256
- Fixed issues with Hover Dynamic DNS #10241
- Updated Cloudflare Dynamic DNS to query Zone ID with token #10992
- Added support for IPv6 to easyDNS Dynamic DNS #10972
- Added support for Domeneshop Dynamic DNS #10826
- Added Zone option to RFC 2136 Dynamic DNS #10684
- Updated FreeDNS Dynamic DNS to use their v2 API #10617
- Fixed DigitalOcean Dynamic DNS processing of zones with multiple pages of records #10592
- Improved Dynamic DNS Logging #10459
- Added support for dynv6.com Dynamic DNS #9642
- Fixed handling of Dynamic DNS AAAA records on 6rd tunnel interfaces bound to PPPoE interfaces #9641

- Added a button to duplicate Dynamic DNS entries #8952
- Fixed Dynamic DNS update for HE.net Tunnelbroker always setting IP address of the default WAN interface #11024
- Updated HE.net Tunnelbroker Dynamic DNS to use their current API #11037
- Added support for Wildcard A records for Gandi Dynamic DNS #11159
- Updated No-IP Dynamic DNS to use a newer API #6638
- Fixed Namecheap Dynamic DNS error code checking #5308
- Improved color blind accessibility of Dynamic DNS status #3229

Gateways

- Added support for obtaining a gateway via DHCP which is outside of the interface subnet #7380
- Added validation to prevent using descriptions on interfaces which would cause gateway names to exceeded the maximum allowed length #9401
- Added tooltip text to icons on the Gateways #10719
- Fixed issues with dpinger failing to update IPv6 gateway address on DHCPv6 WAN interfaces #8136

Hardware / Drivers

- Added `bnxt` driver for Broadcom NetXtreme interfaces #9155
- Added iOS/Android/Generic USB tethering driver #7467

IGMP Proxy

- Added input validation for IGMP Proxy settings #7163

Installer

- Created separate **Auto (UFS) UEFI** and **Auto (UFS) BIOS** installation options to avoid problems on hardware which boots differently on USB and non-USB disks #8638
- Fixed reinstalling with UFS on a ZFS formatted drive #10690
- Fixed platform detection for MBT-4220 and MBT-2220 on newer BIOS revisions #9242
- Fixed an issue with shutting down instead of rebooting after installing using ZFS #7307

Interfaces

- Added support for using IPv4 and IPv6 addresses on GRE interfaces at the same time #10392
- Added a check to disable Hardware Checksum Offloading in environments with interfaces which do not support it (e.g. `vtnet`, `ena`) #10723
- Changed the way interface VLAN support is detected so it does not rely on the VLANMTU flag #9548
- Added a PHP shell playback script `restartallwan` which restarts all WAN-type interfaces #9688

- Changed assignment of the `fe80::1:1` default IPv6 link-local LAN address so it does not remove existing entries, which could cause problems such as Unbound failing to start #9998
- Added automatic MTU adjustment for GRE interfaces using IPsec as a transport #10222
- Fixed SLAAC interface selection when using IPv6 on a link which also uses PPP #9324
- Added GUI interface descriptions to Operating System interfaces #1557
- Added the ability to assign virtual type interfaces (IPsec, OpenVPN, GIF, GRE, etc) during console interface assignment #10947
- Fixed TSO not being disabled in some cases #10836
- Fixed group name length input validation #10835
- Improved interface caching for environments with many interfaces #10680
- Fixed `fe80::1:1` being added to interfaces without track6 #10661
- Added a check to prevent stf (6RD/6to4) interfaces from being used as parent interfaces #10626
- Fixed redundant disabling of static ARP at boot before it could be enabled #10589
- Fixed initialization of bridges which include a GIF interface at boot #10524
- Fixed problems with post-install interface changes not being retained if the user did not complete the wizard #10383
- Fixed inefficiencies when applying settings to a VLAN parent interface #9154
- Fixed interface MTU setting not being applied to all IPv6 routes #6868
- Fixed handling of MTU setting for 6rd and 6to4 interfaces #6377
- Fixed IPv6 IP Alias preventing Track Interface from working with DHCPv6 and RA #5999
- Changed DHCP interface renewal behavior to not restart services if the IP address did not change #11142
- Fixed an error when changing bridge STP settings #11122
- Added a binary package with updated Realtek interface drivers #11079
- Improved link state visibility on Status > Interfaces #11045
- Removed VTI interfaces from Interface Group selection since they do not currently function in this manner #11134
- Fixed issues with IPv6 on top of IPv4 PPPoE placing default route on incorrect interface #9324

IPsec

- Added 25519 curve-based IPsec DH and PFS groups 31 and 32 #9531
- Enabled the strongSwan PKCS#11 plugin #6775
- Added support for ECDSA certificates to IPsec for IKE #4991
- Renamed IPsec "RSA" options to "Certificate" since both RSA and ECDSA certificates are now supported, and it is also easier for users to recognize #9903
- Converted IPsec configuration code from `ipsec.conf` `ipsec/stroke` style to `swanctl.conf` `swanctl/vici` style #9603
  - Split up much of the single large IPsec configuration function into multiple functions as appropriate.

  - Optimized code along the way, including reducing code duplication and finding ways to generalize functions to support future expansion.
  - For IKEv1 and IKEv2 with Split Connections enabled, P2 settings are properly respected for each individual P2, such as separate encryption algorithms #6263
    * **N.B.:** In rare cases this may expose a previous misconfiguration which allowed a Phase 2 SA to connect with improper settings, for example if a required encryption algorithm was enabled on one P2 but not another.
  - New GUI option under **VPN > IPsec**, **Mobile Clients** tab to enable RADIUS Accounting which was previously on by default. This is now disabled by default as RADIUS accounting data will be sent for every tunnel, not only mobile clients, and if the accounting data fails to reach the RADIUS server, tunnels may be disconnected.
  - Additional developer & advanced user notes:
    * For those who may have scripts which touched files in `/var/etc/ipsec`, note that the structure of this directory has changed to the new swanctl layout.
    * Any usage of `/usr/local/sbin/ipsec` or the stroke plugin must also be changed to `/usr/local/sbin/swanctl` and VICI. Note that some commands have no direct equivalents, but the same or better information is available in other ways.
    * IPsec start/stop/reload functions now use `/usr/local/sbin/strongswanrc`
    * IPsec-related functions were converged into `ipsec.inc`, removed from `vpn.inc`, and renamed from `vpn_ipsec_<name>` to `ipsec_<name>`
  - Reworked how reauthentication and rekey behavior functions, giving more control to the user compared to previous options #9983
- Reformatted `status_ipsec.php` to include more available information (rekey timer, encryption key size, IKE SPIs, ports) #9979
- Added support for PKCS#11 authentication (e.g. hardware tokens such as Yubikey) for IPsec #9878
- Fixed usage of Hash Algorithm on child ESP/AH proposals using AEAD ciphers #9726
- Added support for IPsec remote gateway entries using FQDNs which resolve to IPv6 addresses #9405
- Added manual selection of Pseudo-Random Function (PRF) for use with AEAD ciphers #9309
- Added support for using per-user addresses from RADIUS and falling back to a local pool otherwise #8160
- Added an option which allows multiple tunnels to use the same remote peer in certain situations (read warnings on the option before use) #10214
- Improved visible distinction of online/offline mobile IPsec users in the IPsec status and dashboard widget #10340
- Added options to change the IPsec NAT-T ports (local and remote) #10870
- Improved boot-time initialization of IPsec VTI interfaces #10842
- Added support for limiting IPsec VPN access by RADIUS user group #10748
- Changed IPsec to share the same RADIUS Cisco-AVPair parser code as OpenVPN for Xauth users #10469
- Fixed handling of IPsec VTI interfaces in environments with large numbers of IPsec tunnels #9592
- Added IPsec Advanced option to control maximum allowed Parallel P2 Rekey exchanges #9331
- Fixed issues with bringing up new Phase 2 entries on IPsec tunnels with "Split connections" enabled #8472
- Fixed issues where, in rare cases, IPsec tunnels would not reconnect until the firewall was rebooted #8015

- Improved the Remote Gateway field description for IPsec Phase 1 entries to indicate that `0.0.0.0` is allowed #7095
- Fixed issues with IKEv2 IPsec tunnels with multiple phase 2 entries combining traffic selectors in unexpected ways (set "Split Connections" to isolate them) #6324
- Added options to create IPsec bypass rules which prevent specific source and destination network pairs from entering policy-based IPsec tunnels #3329
- Documented settings which work around SA duplication issues experienced by users in certain cases #10176
- Improved IPsec GUI options for P1/P2 SA expiration and replacement to help prevent SA duplication #11219
- Fixed a PHP error in mobile IPsec input validation #11212
- Added validation to prevent unsupported wildcard certificates from being selected for use with IPsec #11297

IPv6 Router Advertisements (RADVD)

- Fixed Router Advertisement configuration missing information in Unmanaged mode #9710
- Fixed Router Advertisement lifetime input validation #10709

L2TP

- Fixed L2TP secret using an empty value after removing it from the GUI #10710
- Fixed L2TP input validation to allow leaving the remote address field blank when assigning addresses from RADIUS #7562
- Fixed inefficiencies in the initial L2TP reconfiguration process #7558
- Fixed L2TP Server and Client both using `l2tpX` for interface names #11006
- Fixed static routes on L2TP interfaces not being reapplied when reconnecting #10407
- Fixed L2TP server being restarted when making user account changes #11059

LAGG Interfaces

- Improved Interface Status and Widget information for LAGG #9187
- Fixed route for GIF/GRE peer when using VLAN on LAGG #10623
- Added option to toggle LACP PDU transmission fast timeout #10504
- Fixed LAGG member interface events causing filter reloads #10365
- Fixed issues with LAGG interface MTU being incorrectly applied to VLAN subinterfaces #8585
- Added option to control the master interface for LAGG in Failover mode #1019

Logging

- Changed system logging to use plain text logging and log rotation, the old binary clog format has been deprecated #8350
- Updated default log size (512k + rotated copies), default lines to display (500, was 50), and max line limits (200k, up from 2k) #9734
- Added log tabs for nginx, userlog, utx/lastlog, and some other previously hidden logs #9714
- Relocated Package Logs into a tab under System Logs and standardized display/filtering of package logs #9714
- Added GUI options to control log rotation #9711
- Added code for packages to set their own log rotation parameters #9712
- Removed the redundant `nginx-error.log` file #7198
- Fixed some instances where logs were mixed into the wrong log files/tabs (Captive Portal/DHCP/squid/php/others) #1375
- Reorganized/restructured several log tabs #9714
- Added a dedicated authentication log #9754
- Added an option for RFC 5424 format log messages which have RFC 3339 timestamps #9808
- Fixed an issue where a firewall log entry for loopback source/destination occasionally reported `127.0.0.1` as `127.0.01` #10776
- Fixed issues with `syslogd` using an old IP address after an interface IP address change #9660
- Added `watchfrr` to routing log #11207

Multi-WAN

- Fixed Gateways being removed from routing groups based on low alert thresholds #10546
- Fixed a possible race condition in gateway group fail-over causing unexpected behavior #9450
- Fixed a load balancing failure when one gateway had a weight of 1 and another gateway had a weight >1 #6025

NAT Reflection

- Fixed port forwards where the destination is a network alias creating invalid refection rules if multiple subnets are in that alias #7614

Notifications

- Deprecated & Removed Growl Notifications #8821
- Added a daily certificate expiration notification with settings to control its behavior #7332
- Fixed input validation of SMTP notification settings #8522
- Added support for sending notifications via Pushover API #10495
- Added support for sending notifications via Telegram #10354
- Fixed a PHP error when SMTP notifications fail #11063

NTPD

- Added GUI options for NTP sync/poll intervals #6787
- Added validation to prevent using `noselect` and `noserve` with pools #9830
- Added feature to automatically detect GPS baud rate #7284
- Fixed status and widget display of long hostnames and stratum #10307
- Fixed handling of the checkbox options on NTP servers #10276
- Updated GPS initialization commands for Garmin devices #10327
- Added an option to limit NTP pool server usage #10323
- Added option to force IPv4/IPv6 DNS resolution for NTP servers #10322
- Added support for NTP server authentication #8794
- Added an option to disable NTP #3567
- Added units to the NTP status page #2850

OpenVPN

- Updated OpenVPN to 2.5.0 #11020
  - The default compression behavior has changed for security reasons. Incoming packets will be decompressed, outgoing packets will not be compressed. There is a GUI control to alter this behavior.
  - Data cipher negotiation (Formerly known as Negotiable Cryptographic Parameters, or NCP) is now compulsory. Disabling negotiation has been deprecated. The option is still present in the GUI, but negotiation will be unilaterally enabled on upgrade. The upgrade process will attempt to use the expected data encryption algorithms before and after the upgrade completes, but in some cases more secure algorithms may be enabled as well. #10919

    We strongly encourage using AEAD ciphers such as AES-GCM, future versions of OpenVPN will require them and will not have configurable cipher lists.
- Added connection count to OpenVPN status and widget #9788
- Enabled the OpenVPN x509-alt-username build option #9884
- Restructured the OpenVPN settings directory layout
  - Changed from `/var/etc/openvpn[-csc]/<mode><id>.<file>` to `/var/etc/openvpn/<mode><id>/<x>`
    - This keeps all settings for each client and server in a clean structure
- Moved to `CApath` style CA structure for OpenVPN CA/CRL usage #9915
- Added support for OCSP verification of client certificates #7767
- Fixed a potential race condition in OpenVPN client ACLs obtained via RADIUS #9206
- Added support for more protocols (IP, ICMP), ports, and a template variable (`{clientip}`) in OpenVPN client ACLs obtained via RADIUS #9206
- Added the ability to register OpenVPN Remote Access (User Auth) clients in the DNS Resolver #10999
- Fixed an issue where duplicating an OpenVPN instance did not copy the password #10703
- Fixed issues with OpenVPN TCP clients failing to start #10650

- Added support for IPv6 OpenVPN ACLs obtained via RADIUS #10454
- Fixed validation to enforce OpenVPN client password usage when setting a username, to prevent a missing password from interrupting the boot process #10409
- Enabled asynchronous push in OpenVPN binary #10273
- Added OpenVPN client-specific override option to ignore routes pushed by the server ("push-reset") #9702
- Clarified behavior of OpenVPN server option for Duplicate Connections #10363

Operating System

- Fixed a network performance regression in the fast forwarding path with IP redirects enabled NG4965
- Fixed double ZFS entries in `loader.conf` #10375
- Added a method to enable persistent command history in the shell #11029
- Changed the default domain name of the firewall from `.localdomain` to `.home.arpa` #10533

Package System

- Disabled spell checking on package upgrade progress textarea #10637
- Fixed issues with package upgrade or reinstall hanging indefinitely #10610
- Fixed description used for buttons when editing packages #11208
- Deprecated the following packages: OpenBGPd, Quagga_OSPF, routed, blinkled, and gwled

PPP Interfaces

- Fixed issues with PPPoE over a VLAN failing to reconnect #9148
- Enabled selection of QinQ interfaces for use with PPP #9472
- Added option to set `Host-Uniq` value for PPPoE #10597
- Fixed incorrect interface assignment after switching from PPPoE #10240
- Fixed IPv6 not being disabled in mpd.conf when the IPv6 GUI option is set to 'disabled' #7386
- Fixed PPPoE interface errors due to MTU settings #11035

PPPoE Server

- Fixed PPPoE server ignoring secondary RADIUS Server #10926
- Fixed PPPoE server Accounting updates option #10869
- Removed unnecessary restarts of the PPPoE server when adding/modifying users #10318
- Added input validation to prevent enabling the PPPoE server on a PPPoE client interface #4510

Routing

- Fixed automatic static routes set for DNS gateway bindings not being removed when no longer necessary #8922
- Fixed missing tooltip text for icons on the Static Routes Page #10889

RRD Graphs

- Fixed RRD graph handling of NTP graph data with negative `freq` values #6503
- Fixed RRD graph creation for interfaces using CODELQ #6277

Rules / NAT

- Added the ability to configure negated tagging, to match packets which do not not contain a given tag #10186
- Added support for IPv6 Port Forwards #10984
- Fixed handling of IPv6 NPt rules on 6rd WAN interfaces #10757
- Fixed 1:1 NAT issue when internal interface has VIPs #10752
- Fixed policy routing rules not being written correctly for a down gateway #10716
- Added EoIP to firewall rule Protocol list #10698
- Fixed separator bars on floating rules not covering the full table width #10667
- Fixed 1:1 NAT for IPv6 applying wrong subnet mask to "Single Host" #7742
- Added validation to prevent accidentally overlapping NPt networks and interface networks #7741
- Added support for dynamic interface addresses in 1:1 NAT rules #7705
- Added default values of TCP and UDP timeouts to the GUI #7362
- Fixed handling of IPv6 floating rules on 6rd interfaces #7142
- Fixed firewall rules for "PPPoE clients" only including the first PPPoE server instance #6598
- Fixed duplicated tracker IDs on block private networks rules #6030
- Fixed `reply-to` on rules for PPPoE WANs with IPv6 SLAAC #5258
- Added gateway/group IP addresses to mouseover on rules #885
- Fixed formatting of floating rules with large numbers interfaces #10892
- Fixed form rendering issues with Port Forward Address Fields in Safari #10674
- Fixed firewall ruleset failing to load at boot when new ruleset would be invalid #6028
- Fixed an issue adding or deleting separator bars when no rules are present #10827

S.M.A.R.T.

- Updated S.M.A.R.T. Page with new capabilities #9367

SNMP

- Fixed SNMP reporting incorrect speed for switch uplink interface on Netgate SG-3100 #10793
- Fixed SNMP input validation to require the Host Resources module when the PF module is also enabled #10471

Traffic Graphs

- Changed the Traffic Graph page from `rate` to `iftop` which brings IPv6 support and various other improvements #3334

Traffic Shaper (ALTQ)

- Changed default ALTQ queue bandwidth type to Mbit/s #10988
- Updated traffic shaper wizard settings for XBox and Wii ports #10837
- Added Broadcom NetXtreme to ALTQ-capable list #10762
- Added ALTQ support to the `ix(4)` driver #7378
- Fixed deletion of associated shaper queues when deleting an interface #3488
- Fixed ALTQ root queue bandwidth calculation #3381
- Fixed input validation for amount of queues supported by ALTQ schedulers #1353
- Added Google Stadia port range to the traffic shaper wizard #10743
- Fixed PHP errors in the traffic shaper wizard #10660
- Fixed ALTQ on `hn(4)` interfaces #8954

Traffic Shaper (Limiters)

- Fixed issues with `net.inet.ip.dummynet.*` tunables being ignored #10780
- Fixed issues with renaming limiters removing them from firewall rules #3924
- Fixed mask options not applying to sched limiter #10838
- Changed default Limiter queue bandwidth type to Mbit/s #10727

Translations

- Added Italian translation #9716

Upgrade

- Fixed issues with checking for updates from the GUI behind a proxy with authentication #9478
- Changed phrasing of message indicating the firewall is rebooting to upgrade #10387
- Fixed issues with the GUI incorrectly reporting "The system is on the latest version" #8870

UPnP

- Improved handling of UPnP with multiple gaming systems #7727

User Manager / Privileges

- Added menu entry for User Password Manager if the user does not have permission to reach the User Manager #9428
- Improved consistency of SSL/TLS references in LDAP authentication servers #10172
- Fixed irrelevant output being printed to users with `ssh_tunnel_shell` #9260
- Fixed theme not being applied to LDAP test results modal #7912
- Changed to more secure default values for certificates created through the user manager #11167
- Changed SSL/TLS LDAP authentication implementation to improve handling of multiple secure LDAP (SSL/TLS or STARTTLS) servers used at the same time #10704

Virtual IP Addresses

- Fixed a problem with PID file handling for the proxy ARP daemon #7379
- Fixed IP Alias VIPs on PPPoE interfaces #7132

Web Interface

- Updated JQuery to address multiple issues #10676
- Updated Bootstrap to 3.4.1 #9892
- Updated Font-Awesome to v5 #9052
- Increased the number of colors available for the login screen #9706
- Added TLS 1.3 to GUI and Captive Portal web server configuration, and removed older versions (TLS 1.0 removed from Captive Portal, TLS 1.1 removed from GUI) #9607
- Fixed empty lines in various forms throughout the GUI #9449
- Improved validation of FQDNs #9023
- Added `CHACHA20-POLY1305` to `nginx` cipher list #9896
- Fixed Setup Wizard input validation to allow Primary/Secondary DNS Server field to remain empty #10982
- Fixed Setup Wizard input validation for IPv6 DNS Servers #10720
- Added an option to omit DNS Servers from resolv.conf #10931
- Fixed the icon area within buttons not being clickable #10846

- Fixed visibility issues with multiple selection form control in the `pfsense-BETA-dark` theme #10705
- Updated documentation links in the GUI #10481
- Fixed netmask/prefix form control incorrectly resetting to `128/32` #10433
- Updated Help shortcut links #10135
- Improved handling of multiple login form submissions to avoid a potential CSRF error #9855
- Fixed reboot message when changing the Hardware Checksum Offloading setting #3031
- Added support for new site icons requested by current versions of Safari #11068
- Added descriptions to all `write_config()` calls #204

WireGuard

- Added kernel-based *WireGuard* VPN implementation #8786

Wireless

- Added support for the `athp(4)` wireless interface driver #9538 #9600
- Added support for the `ral(4)` wireless interface driver to arm64 #10934
- Added support for the `rtwn(4)` wireless interface driver #10639
- Added support for selecting 802.11n channel width (HT) #10678

Development

- Added a "periodic" style framework to allow for daily/weekly/monthly tasks from the base system or packages by way of plugin calls #7332
- Added a central file download function for internal use throughout the GUI
- Added TCP_RFC7413 in kernel, required for the BIND package #7293

XMLRPC

- Fixed XMLRPC synchronization of admin authorized keys for the admin user #9539
- Added option to synchronize changes for the account used for XMLRPC sync #9622
- Fixed XMLRPC synchronization for firewall rule descriptions with special characters #1478
- Fixed Incorrect synchronize IP address value causing XMLRPC errors #11017

3.2.7 2.4.5-p1 New Features and Changes

pfSense® software version 2.4.5-p1 addresses performance, security, and other miscellaneous issues found in *2.4.5*.

Warning: Proceed with caution when upgrading pfSense software while COVID-19 travel restrictions are in effect.

During this time of travel limitations, remote upgrades of pfSense software should be carefully considered, and avoided where possible. Travel restrictions may complicate any repair of any issue, including hardware-related issues that render the system unreachable. Should these issues require onsite physical access to remedy, repair of the issue may not be possible while travel restrictions related to COVID-19 are in effect.

Tip: For those who have not yet updated to 2.4.4-p3 or 2.4.4, consult the *previous release notes* and blog posts for those releases to read all important information and warnings before proceeding.

Note: Upgrading to pfSense software version 2.4.5-p1 requires `pfSense-upgrade` version 0.70 or later. Most installations will automatically pick up the new version and upgrade normally. If this does not happen automatically and the upgrade to version 2.4.5-p1 is not offered, use the following procedure:

- Navigate to **System > Updates**
- Set **Branch** to *Previous stable version*
- Wait a few moments for the upgrade check to complete
- Optional: Confirm that the latest version of *pfSense-upgrade* is present (version >= 0.70) using `pkg-static info -x pfSense-upgrade`.

  If the correct version is not present, wait a bit longer and check again as that package may be updating in the background.
- Set **Branch** to *Latest stable version*
- Wait a few moments for the upgrade check to complete

At this point, the upgrade check should see 2.4.5-p1 and the upgrade can proceed.

Note: pfSense software version 2.4.5-p1 includes `pkg` version 1.13.x which introduces a new metadata version. Most installations will automatically pick up the new version and upgrade normally. In certain cases, especially coming from much older versions, the `pkg` utility may require a manual update before it can correctly process the new metadata.

The `pkg` utility can be upgraded manually with the following command run from an ssh or console shell:

```
# pkg-static bootstrap -f
```

See *Repository Metadata Version Errors* for more details.

Security / Errata

- Addressed an issue with large `pf` tables causing system instability and high CPU usage during filter reload events #10414
- Fixed an issue with `sshguard` which could prevent it from protecting against brute force logins #10488
- Updated `unbound` to address CVE-2020-12662 and CVE-2020-12663 #10576
- Updated `json-c` to address CVE-2020-12762 #10609
- Addressed FreeBSD Security Advisories & Errata Notices including:
  - FreeBSD-SA-20:10.ipfw
  - FreeBSD-SA-20:12.libalias
  - FreeBSD-SA-20:13.libalias
  - FreeBSD-SA-20:15.cryptodev

Aliases / Tables

- Fixed handling of URL/URL table aliases with IDN hostnames #10321

Authentication

- Fixed handling of misconfigured groups which prevented the `admin` user from making configuration changes #10492
- Fixed a potential temporary privilege downgrade when deleting an account #9259

Backup / Restore

- Fixed handling of redundant/extraneous RRD tags when making configuration backups #10508

CARP

- Fixed handling of IPv6 CARP VIPs with non-significant zeros during XMLRPC sync #6579

Certificates

- Fixed a bug which prevented the user from removing a CA private key when editing #10509

Configuration Upgrade

- Fixed a PHP error during upgrade from <2.4.3 with empty tags in the IPsec configuration #10458

Console Menu

- Changed the naming convention of gateways created at the console to be the same as those created in the GUI #10264

DHCP (IPv6)

- Added default value placeholders to some DHCPv6 RA configuration options #10448
- Fixed DHCPv6 service Dynamic DNS errors #10346
- Fixed `rc.newwanipv6` being called for Request messages which `dhcp6c` should have discarded #9634
- Added dashed DUID support to DHCPv6 static mappings #2568

DHCP Relay

- Fixed DHCP Relay handling of scenarios where a target server may be on the same interface as some clients #10416
- Excluded unsupported interface types from DHCP Relay #10341

DHCP Server

- Fixed DHCPv6 static entries not being updated on external Dynamic DNS servers #10412
- Fixed DHCPv6 `domain-search` list not being sent to clients #10200
- Fixed DHCP Server not accepting IPv6 addresses for Dynamic DNS servers #6600

Diagnostics

- Several improvements and items added to status.php diagnostic output #10455 #10424 #10423 #10350 #10349 #10568
- Fixed Require State Filter setting on `diag_states.php` breaking filter rule link to associated states #10359

DNS Resolver

- Fixed IPsec and OpenVPN IPv6 tunnel network/pool prefixes not being added to automatic DNS Resolver ACLs #10460
- Fixed EDNS buffer size values to prepare for 2020 DNS flag day #10293
- Fixed DNS Resolver handling of entries from DHCP server which contain a trailing dot in domain names #8054

Dynamic DNS

- Fixed DigitalOcean Dynamic DNS client handling of IPv6 addresses #10390
- Fixed DNSExit update URL #9632

Hardware / Drivers

- Added support for `iwm` devices #7725

> **Note:** This device only supports Station mode. It does not support acting as an access point.

- Added `ng_etf` module to `armv6` and `aarch64` kernels #10463
- Added QLogic 10G driver (`qlxgb`/`qla80xx`) #9891
- Added `virtio_console` to the kernel #9985

IPsec

- Fixed selection of IPsec VTI Phase 2 local network address/mask values #10418
- Fixed saving IPsec connection breaking FRR BGP on VTI interfaces #10351
- Updated DH group warnings to say that group 5 is also weak #10221
- Fixed disabling IPsec Phase 1 with a VTI Phase 2 #10190
- Fixed disabled IPsec Phase 2 entries being unintentionally included in `vpn_networks` table #7622

L2TP

- Changed L2TP `mpd.secret` handling so that the server is not restarted after adding/modifying L2TP users #4866
- Fixed handling of L2TP usernames containing a realm separator (`@`) #9828
- Fixed Shared Secret handling in L2TP #10531 #10527

Limiters

- Fixed input validation of limiters with ECN #10211
- Fixed bogus extra warning dialog on when deleting limiters #9334

Notifications

- Fixed SMTP notification SSL validation to respect the user-selected behavior #10317

NTPD

- Added `localhost` to NTP Interface selection options #10348

OpenVPN

- Fixed OpenVPN `remote` statement protocol handling #10368
- Added option to configure OpenVPN username as common name behavior #8289

Operating System

- Fixed handling of RAM disk sizes not accounting for existing disk usage when calculating available kernel memory, which could prevent saving #10420
- Updated `pkg` to 1.13.x #10564
- Fixed problems preventing the Netgate Coreboot Package from updating Coreboot properly #10573

Packages

- Fixed handling of FreeRADIUS passwords containing non-XML-safe characters #4497
- Fixed handling of Squid LDAP search filters containing an accent #7654
- Fixed issues preventing FRR from working on certain platforms such as SG-1100 (arm64/aarch64) #10444
- Fixed issues preventing Suricata from working on certain platforms such as SG-1100 (arm64/aarch64) #10228

Rules / NAT

- Fixed Duplicate Outbound NAT entries from L2TP server addresses #10247
- Fixed Outbound NAT rules for mobile IPsec users with per-user addresses defined #9320
- Fixed IPv6 IP Alias VIPs not being added to Interface Network macros #8256
- Fixed Destination port range “Any” in Port Forward rules #7704
- Fixed display of interfaces on the Floating rules list #4629
- Fixed rule description validation to reject \ #10542
- Fixed setting NAT reflection timeout values #10591

Translations

- Fixed language selection for Chinese (Taiwan) / HK Translations #10525

Services

- Fixed `is_process_running()` handling of empty process, which could lead to an error when using the CLI to query the status of a service which does not exist #10540

Web Interface

- Fixed dark theme auto-complete popup field having dark text on dark background #10499
- Fixed using special characters in Schedule descriptions #10305
- Fixed WebGUI main page loading very slowly when there is no Internet connectivity #8987

3.3 Older Unsupported Releases

3.3.1 2.4.5 New Features and Changes

pfSense® software version 2.4.5 contains a variety of bug fixes and maintenance updates.

Warning: Proceed with caution when upgrading pfSense software while COVID-19 travel restrictions are in effect.

During this time of travel limitations, remote upgrades of pfSense software should be carefully considered, and avoided where possible. Travel restrictions may complicate any repair of any issue, including hardware-related issues that render the system unreachable. Should these issues require onsite physical access to remedy, repair of the issue may not be possible while travel restrictions related to COVID-19 are in effect.

Tip: For those who have not yet updated to 2.4.4-p3 or 2.4.4, consult the *previous release notes* and blog posts for those releases to read all important information and warnings before proceeding.

Operating System / Architecture changes

- Base OS upgraded to FreeBSD 11.3-STABLE@r357046
- PHP upgraded to 7.2.29

Security / Errata

- Fixed dependency issues with `pfSense-upgrade` which may have caused it not to update itself properly #10303

> **Tip:** If the update check fails, or the update does not complete, run `pkg install -y pfSense-upgrade` to ensure that `pfSense-upgrade` is present.

- Added encoding to the hostname in `services_acb.php` #9584
- Added encoding to error output in `services_captiveportal_mac.php` #9609
- Improved Picture Widget input validation #9610 #9731 #9804
- Added a `fsck` run with `-z` for `UFS` filesystems on upgrade to address FreeBSD-SA-19:10.ufs #9612
- Fixed format of XMLRPC auth error to match GUI auth error #9782
- Added a custom CSRF Error page with warnings and confirmation prompts before resubmitting potentially harmful data #9799
- Fixed Status_Monitoring `rrd_fetch_json.php` error encoding #9601
- Fixed encoding of the user full name on `system_usermanager_addprivs.php` #10324
- Fixed input validation and output encoding of host on `diag_ping.php` #10355
- Addressed FreeBSD Security Advisories & Errata Notices
  - FreeBSD-SA-20:05.if_oce_ioctl
  - FreeBSD-SA-20:04.tcp
  - FreeBSD-SA-19:24.mqueuefs
  - FreeBSD-SA-19:23.midi
  - FreeBSD-SA-19:22.mbuf
  - FreeBSD-SA-19:21.bhyve
  - FreeBSD-SA-19:20.bsnmp
  - FreeBSD-SA-19:19.mldv2
  - FreeBSD-SA-19:18.bzip2
  - FreeBSD-SA-19:17.fd
  - FreeBSD-SA-19:16.bhyve
  - FreeBSD-SA-19:15.mqueuefs
  - FreeBSD-SA-19:14.freebsd32
  - FreeBSD-SA-19:13.pts
  - FreeBSD-SA-19:12.telnet
  - FreeBSD-SA-19:11.cd_ioctl
  - FreeBSD-SA-19:10.ufs
  - FreeBSD-SA-19:09.iconv
  - FreeBSD-SA-19:08.rack

- FreeBSD-EN-20:06.ipv6
- FreeBSD-EN-20:04.pfctl
- FreeBSD-EN-19:18.tzdata
- FreeBSD-EN-19:17.ipfw
- FreeBSD-EN-19:16.bhyve
- FreeBSD-EN-19:15.libunwind
- FreeBSD-EN-19:14.epoch
- FreeBSD-EN-19:13.mds
- FreeBSD-EN-19:12.tzdata
- FreeBSD-EN-19:11.net

Aliases/Tables

- Fixed an issue when resolving FQDN entries in aliases where some entries could be missing #9296
- Improved URL Table aliases to support FQDNs which return muliple entries #8531
- Added a function to download the contents of an individual alias #9816

Authentication

- Added exception handling to authentication attempts #9150

Backup/Restore

- Added a special string (`NoReMoTeBaCkUp`) that when used in `write_config()` descriptions will prevent a remote backup #9693
- Removed legacy AutoConfigBackup options (there were no more active accounts using the retired legacy service) #9687 #9785
- Added CDATA protection to the `encryption_password` XML tag, which allows international characters to be used in that field #7186
- Added CDATA escape to more auth-related fields #9327
- Ensured that `kern.cam.boot_delay` is set for new installations and upgrades so that USB devices are properly initialized in time for configuration restore in the installer and ECL to function #9533

Captive Portal

- Fixed Captive Portal vouchers shortcut links #9722
- Changed Captive Portal redirect page selection order #9819
- Fixed a rare and intermittent issue where users could encounter an `nginx` error when restarting Captive Portal instances #10159

Certificates

- Added sorting and search/filtering to Certificate Authority & Certificate manager #9412
- Corrected wording of CA/Cert CN input validation #9234
- Fixed certificate Descriptive Name field behavior when adding a user certificate #9719
- Added `clientAuth` EKU to Server type certificates #9868
- Reduced the default GUI web server certificate lifetime to 398 days to prevent errors on Apple platforms #9825

Dashboard

- Added option to disable PTI display in System Information widget #9323

DHCP

- Fixed incorrect expansion of Dynamic DNS advanced options on the DHCPv6 Server page #9448
- Changed DHCP relay backend code to determine and specify separate upstream and downstream interface lists #9466
- Prevented OpenVPN interfaces from being used by DHCP relay, since that type of interface is not compatible #8443
- Added an option to disable ping check in dhcpd #9285
- Fixed **Show all configured leases** so it is persistent after deleting a DHCP lease #9133
- Added search/filter to DHCP/DHCPv6 leases #9791
- Improved DHCP client handling of timeout conditions and script failures #9267

Diagnostics

- Fixed a PHP warning in diag_dump_states.php #9780
- Fixed reverse lookup of IPv6 addresses on diag_dns.php #9543
- Fixed diag_system_activity.php to use batch mode for top so it displays process list w/o terminal, and increased amount of output displayed #9522
- Added search/filter ARP table and NDP status #9791

DNS

- Added `127.0.0.0/8` to the DNS Resolver `private-address` list for DNS rebinding protection #9708
- Fixed CIDR selection issues with /32 entries in DNS Resolver Access List entries #9586
- Fixed an issue saving DNS over TLS hostnames on systems with only one gateway #9898
- Fixed an issue where manually configured DNS servers may not have been active if "allow override" was disabled and they were also assigned dynamically #9963
- Added DNS Resolver (Unbound) Python Integration #9251

Dynamic DNS

- Fixed Dynamic DNS class constructor name #9779
- Fixed errors in DNSimple Dynamic DNS #9580
- Fixed handling of wildcard (`*`) hostname entries in Cloudflare Dynamic DNS #9361
- Added support for AAAA records to Digital Ocean Dynamic DNS #9280
- Fixed issues with Digital Ocean Dynamic DNS handling of empty hostnames #9602
- Cleaned up whitespace issues in Azure Dynamic DNS backend code #9271
- Added support for Linode Dynamic DNS #9268
- Fixed issues with IPv6 on Azure Dynamic DNS #9248
- Fixed handling of wildcards in Route53 Dynamic DNS #9053
- Fixed handling of wildcards in Loopia Dynamic DNS #8014
- Fixed CloudFlare Dynamic DNS processing when `proxied` is enabled #9362
- Fixed CloudFlare Dynamic DNS "Invalid TTL" error due to CloudFlare API update #10196
- Changed hostname to optional for DNS-O-Matic Dynamic DNS #7601
- Added support for Gandi LiveDNS Dynamic DNS #9452

Gateways

- Corrected PHP errors when marking gateways down in certain edge cases #9851

Interfaces

- Added more prefix delegation size entries to selection list on interfaces.php #9590
- Added initialization to the VLAN array in console setup #9582
- Fixed issues with Netgate & hardware model detection which caused problems with default interface mappings #8051
- Fixed issues with display of previously-entered IP address values on interfaces_ppps_edit.php #9741
- Added a confirmation prompt to disconnect/release actions on status_interfaces.php #9911
- Added drivers for Mellanox `mlx4` and `mlx5` network interface cards #7537

IPsec

- Fixed IPsec VTI interface creation logic #9781
- Added GUI option for IPsec P2/Child SA close action #9767
- Added IPsec DH and PFS groups 25, 26, and 27 #9757
- Added 25519 curve-based IPsec DH and PFS group 31 #9531
- Enabled NAT-T controls for IKEv2 #9695
- Improved handling of IPsec restarts breaking VTI routing #9668
- Fixed input validation that incorrectly prevented deleting IPsec P2 entries in some cases with VTI #9258

- Fixed IPsec `keyid` identifier handling #9243
- Fixed IPsec VTI MTU boot-time configuration #9111
- Escape Windows domain backslash in IPsec widget #9747
- Fixed VTI IPv6 address handling #9801
- Fixed Child SA button JS hide on status_ipsec.php, along with other cosmetic improvements #8847
- Added **Connect Children** button to status_ipsec.php to connect when IKE (Phase 1) is up but Child SAs (Phase 2 entries) are not #9954
- Fixed IPsec Phase 2 Remote Network field show/hide when changing between Phase 2 modes #9720
- Fixed IPsec configuration generation so that encryption options for every P2 on a given P1 are not duplicated on each P2 #6263
- Fixed a PHP error in IPsec package plugin hook processing #10217

Load Balancer

- Fixed a PHP when processing services when the configuration does not contain Load Balancer entries #10308

Logging

- Moved `igmpproxy` logs to `routing.log` #10139
- Moved `igmpproxy` verbose logging option to `services_igmpproxy.php` (formerly at `status_logs_settings.php`) #10139
- Updated `sshguard` and fixed a log processing regression #9971
- Fixed PHP errors in filter log processing when entries contain an invalid port #10255

Monitoring

- Fixed custom view titles being forced to lower case #9681
- Fixed packet graph scaling #9807
- Fixed a PHP error in RRD processing of ALTQ data #10248

Notifications

- Fixed SMTP notification password being unintentionally changed when testing SMTP settings #9684
- Reduced frequency of GEOM rebuild notifications #9256

NTPD

- Added validation to ensure NTP values are treated as numbers before use #9558
- Changed the default NTP pool server to `2.<domain>` so that it can use IPv6 #9931
- Improved handling of errors on the NTP status page to work/fail gracefully with custom ACLs for localhost in place #9829

OpenVPN

- Fixed JavaScript issue when selecting multiple OpenVPN NCP algorithms #9756
- Fixed OpenVPN wizard so it does not show DH parameter lengths that are not available #9748
- Fixed issues with OpenVPN resynchronizing when running on a gateway group #9595
- Added an option to set the OpenVPN TLS Key Direction #9030
- Added GUI options to configure OpenVPN keepalive parameters #3473
- Fixed instances of hidden invalid OpenVPN options affecting save operations #9674
- Added a copy action to OpenVPN pages #5851
- Improved sorting of bytes sent/receives on OpenVPN status page #7359
- Fixed visibility of the OpenVPN 'interface' option when multihome is selected #7840
- Reduced the OpenVPN server certificate lifetime to 398 days in the wizard to prevent errors on Apple platforms #9825
- Added input validation to prevent OpenVPN tunnel network reuse #3244
- Added Exit Notify to OpenVPN servers/client options #9078

Operating System

- Fixed serial console terminal size issues #9569
- Added the `strings` binary to base builds for troubleshooting #7791
- Changed UFS filesystem defaults to `noatime` on new installations #9483
- Fixed an issue where the IP header checksum was incorrect when reassembling packet fragments to a link with a different MTU #10189

Packet Capture

- Changed Packet Capture GUI to allow multiple TCP/UDP ports to be specified #9766
- Added start time to Packet Capture display #9831
- Added OSPF/OSPFv3 to Packet Capture protocols #9905
- Fixed Packet Capture to match both IPv4+IPv6 CARP when that protocol is selected #9867
- Fixed Packet Capture for the `pfsync` protocol #10183

Routing

- Fixed `(Default)` designation on routes to match the default route in the OS #9292
- Fixed static routes remaining in routing table after removal #9969

Rules / NAT

- Fixed state kill ordering in rc.newwanip #4674
- Added the ability to search firewall logs by tracking ID #8703
- Added GUI option to disable default blocking of APIPA networks #9966
- Added more common ports to the firewall rule drop-down list #10166
- Added input validation to prevent selecting `!*` ("not any") in source or destination #10168
- Fixed invalid rules generated when using NAT reflection with a negated destination #10246

S.M.A.R.T.

- Updated the SMART page with new capabilities #9367

SNMP

- Fixed SNMP sysDescr contents to include hostname and patch version #9218

Traffic Shaping / Limiters

- Added input validation for Limiter delay values #9921
- Fixed the queue statistics parser to handle large values #9938

Translations

- Fixed an issue with international characters in configuration descriptions, which led to failures in certain cases, such as failing to set Manual Outbound NAT when the Language was set to pt_BR #6195
- Fixed a PHP error on `system_advanced_admin.php` when the language was set to French #10331

Upgrade / Installation

- Revised update check to provide a more consistent version string in JSON format #9778
- Disabled serial console on VGA memstick images #9488
- Fixed a PHP error when upgrading older configurations from revision 14.4 to 14.5 #9840

UPnP

- Fixed display of active UPnP sessions when configured with an alternate external address #9961

User Manager / Privileges

- Added input validation to prevent changing the authentication server name #9692
- Added privilege to manage integrated switches #9620
- Fixed privilege matching to handle JS anchor links #9550
- Removed wildcards incorrectly used in `isAllowedPage()` #9541
  - This issue could prevent a user in the admins group from reaching certain pages such as the User Manager.
- Improved Deny Config Write privilege handling in the User & Group Manager #9259
- Fixed input validation of group name sizes to allow longer remote groups #3792
- Fixed handling of L2TP and PPPoE user passwords containing invalid characters #10275

Web Interface

- Corrected input validation for firewall rule VLAN priority/set #9763
- Restricted Thoth tests to arm64 in status.php NG 2569
- Added kernel memory usage to status.php output #9705
- Redacted several additional fields in status.php output #9784 #9729 #9728 #9727 #9694 #9736 #9764
- Fixed a potential source of PHP errors when saving per-log settings #9540
- Added GUI components for MDS mitigation #9532
- Fixed integrated switch LAGG member editing on switch_ports.php #9447
- Fixed wizard.php selection option size attribute handling #8907
- Fixed platform detection for certain C2558/C2758 systems #6846
- Set `autocomplete=new-password` for forms containing authentication fields to help prevent browser auto-fill from completing irrelevant fields #9864
- Fixed processing of shortcuts for XML-based packages #9770
- Updated jQuery #9407
- Improved consistency of SSL/TLS references throughout the GUI #10172
- Updated various help references and links to use the pfSense book instead of external resources #10135 #10184

XMLRPC

- Fixed removal of the last ALTQ traffic shaping entry from the target system when performing an XMLRPC sync #9469
- Fixed removal of the last limiter entry from the target system when performing an XMLRPC sync #9468

3.3.2 2.4.4-p3 New Features and Changes

pfSense® software version 2.4.4-p3 addresses security and other issues found in *2.4.4-p2*.

Tip: For those who have not yet updated to 2.4.4-p2 or 2.4.4, consult the *previous release notes* and blog posts for those releases to read all important information and warnings before proceeding.

Warning: The upcoming pfSense release version 2.5.0 deprecates the built-in load balancer, and all related code has been removed as it is not compatible with FreeBSD 12. Plan migrations to alternate solutions such as the HAProxy package now.

See the *2.5.0 release notes* for more information.

Security / Errata

- Changed `sshguard` to block both ssh and the GUI using a single table, and removed the unnecessary manual scheduled table expiration pfSense-SA-19_02.sshguard #9223
- Fixed potential XSS vectors
  - pfSense-SA-19_01.webgui : Fixed potential XSS vectors in system_advanced_admin.php, interfaces_assign.php, firewall_rules_edit.php, firewall_shaper.php, services_igmpproxy_edit.php, services_ntpd_gps.php and diag_traceroute.php #9294
  - pfSense-SA-19_03.webgui : Fixed potential XSS vector in status_filter_reload.php #9499
  - pfSense-SA-19_04.webgui : Fixed potential XSS vector in the WOL widget #9507
  - pfSense-SA-19_05.webgui : Fixed potential XSS vector in services_acb.php #9508
- Fixed privilege issues
  - pfSense-SA-19_06.webgui : Restrict edit access to OpenVPN-related advanced settings, and added new privilege to delegate edit permissions #9511
  - pfSense-SA-19_07.webgui : Strengthen widget privilege matching to avoid a potential privilege bypass for users granted access to widgets #9512
  - pfSense-SA-19_08.webgui : Strengthen path privilege check to avoid a potential directory-traversal-like bypass method #9513
  - Added privileges for Auto Config Backup pages #9519
  - Updated privileges: Added misc missing pages, removed obsolete pages
- Addressed FreeBSD Security Advisories:
  - FreeBSD-SA-19:03.wpa
  - FreeBSD-SA-19:04.ntp

  - FreeBSD-SA-19:05.pf
  - FreeBSD-SA-19:06.pf
  - FreeBSD-SA-19:07.mds
  - FreeBSD-EN-19:08.tzdata
- Added DNS over TLS host verification #8602
  - Configure hostnames for DNS over TLS servers under **System > General**
- sqlite updates #9205

Backup / Restore

- Fixed issues with output buffering causing configuration backup download failures #9390
- Fixed automatic package reinstallation after restoring config.xml from the installer #9214
- Force `<enableserial>` when restoring a backup on a device with serial only console

Certificates

- Added missing countries from CA list on certificate pages #9308
- Fixed an error when adding a new user and choosing to generate a certificate #9317

DNS

- Fixed input validation on diag_dns.php to allow a trailing dot on hostnames #9276
- Removed non-functional tools links from diag_dns.php #9275
- Fixed rewriting of the DNS Resolver file `remotecontrol.conf` if it is present but empty #9470

Firewall Rules / NAT / Aliases

- Fixed intermittent pf errors when NAT reflection is enabled #9446
- Fixed reserved pf keyword matching when creating and editing aliases #9231
- Fixed duplicate entries showing on diag_tables.php from lockout tables #9359
- Fixed a PHP error deleting an imported NAT rule with no firewall rules present #9193
- Do not show scheduler icon when scheduler tag is empty

Gateways / Routing

- Fixed issues with the default IPv4 gateway set to a group failing after restart #9004

Interfaces

- Fixed PHP error from interface groups when editing QinQ entries

IPsec

- Fixed IPsec Phase 1 entries on upgrade to have their `protocol` field populated properly #9207

Operating System

- Fixed support for ZFS encrypted+mirrored swap #9281
- Fixed problems saving crash dumps when `/var` is a RAM disk #9409

Traffic Shaping

- Fixed a PHP error when loading a limiter that does not exist #9313
- Fixed limiter selection validation
- Fixed Queues menu items ending with ":" in certain languages #8970

WebGUI

- Numerous optimizations and improvements for status.php diagnostics output #9290
- Fixed a PHP error on system_advanced_network.php when disabling "IPv6 over IPv4 Tunneling" #9264
- Improved handling of large captures on diag_packet_capture.php and disabled viewing of captures larger than 50MiB. #9239
- Added hostname to login page title if the user has enabled **Show hostname on login banner** #9096
- Centralized the list of country codes used by multiple areas #9308
- Updated translation files

XMLRPC

- Clarified conditions for synchronizing certificates in HA Sync options #9283

3.3.3 2.4.4-p2 New Features and Changes

pfSense® software version 2.4.4-p2 adds support for new Netgate hardware and corrects issues found with *2.4.4-p1*.

> **Warning:** For those who have not yet updated to 2.4.4-p1 or 2.4.4, consult the release notes and blog posts for those releases to read all important information and warnings before proceeding.

Miscellaneous

- Hardware support/improvements for Netgate products
- Fixed swap slice labeling in MBR mode and changed the way swap is located at boot time to detect and work around incorrect `fstab` swap labels created by the installer #9182
- Fixed handling of IPv6 name servers with nginx when using a certificate that requires OCSP stapling #9160
- Fixed handling of NPt rules using a /128 prefix #9163
- Fixed a PHP error in the Setup Wizard when dealing with static gateways #9170
- Updated Dynamic DNS to accommodate recent changes in the Digital Ocean API #9171
- Fixed OpenVPN RADIUS authentication use of `calling_station_id` #9178
- Fixed input validation that rejected certain valid hash algorithms when signing a CSR #9180
- Removed obsolete and unused OLSRD code #9117

3.3.4 2.4.4-p1 New Features and Changes

pfSense® software version 2.4.4-p1 corrects issues found with 2.4.4-RELEASE.

Security / Errata

- FreeBSD Errata Notice FreeBSD-EN-18:09.ip: IP fragment remediation causes IPv6 fragment reassembly failure #8934
- FreeBSD Errata Notice FreeBSD-EN-18:10.syscall NULL pointer dereference in freebsd4_getfsstat system call (CVE-2018-17154)
- FreeBSD Errata Notice FreeBSD-EN-18:11.listen Denial of service in listen syscall over IPv6 socket (CVE-2018-6925)
- FreeBSD Errata Notice FreeBSD-EN-18:12.mem Small kernel memory disclosures in two system calls (CVE-2018-17155)
- Fixed a potential authenticated command injection issue with PowerD settings pfSense-SA-18_09.webgui #9061
- Fixed handling of privileges on the **All** group that were previously ignored #9051

> **Warning:** Check the privileges on the **All** group before upgrading to avoid unintended privileges for accounts being respected that were not honored before

Certificates

- Fixed CRL lifetime errors due to 2038 rollover on 32-bit ARM platforms #9098
- Fixed date display of CA/Certificate validity ending dates after 2038 rollover on 32-bit ARM platforms #9100
- Fixed PHP errors when creating certificate entries #9099

DNS

- Updated Unbound to 1.8.1 to address issues with memory leaks, especially in DNS over TLS support #9059
- Fixed issues with the DNS search domain for the firewall being omitted from `resolv.conf` in certain cases #9056
- Fixed PHP errors in the DNS Forwarder #8967

Dynamic DNS

- Fixed an issue with FreeDNS Dynamic DNS sending an IP address with an update #8924
- Fixed issues with Custom (v6) Dynamic DNS logging a hostname error #8977

DHCP Server

- Fixed issues with DHCPv6 network boot settings #8949

Routing/Gateways

- Reduced the logging output of gateway change events #8914
- Fixed an issue with `dpinger` PID files causing it to get stuck in Pending status #8921
- Fixed display of a configured gateway monitor IP address when gateway monitoring is disabled #8953
- Fixed issues with double quotes in gateway descriptions causing a blank gateway drop-down on firewall rules #8962
- Fixed an issue where the default gateway was lost in certain cases with HA after a CARP VIP status transition #8465

IPsec

- Updated strongSwan to 5.7.1 #8898
- Added `0.0.0.0/0` to both sides of an IPsec VTI P2 to allow connections with third-party routed IPsec implementations that require its presence #8859
- Fixed boot-time handling of IPsec VTI static routes #9116
- Fixed IKEv2 EAP Identity/Client ID matching so that it is strictly performed, to avoid users getting incorrect per-user settings #9055
- Fixed handling of RADIUS server names containing a `.` in the IPsec configuration with strongSwan 5.7.1 #9106
- Updated AWS IPsec wizard to use EC2 instance profiles and security groups, and switched the wizard from OpenBGPD to FRR

Interfaces/VIPs

- Fixed issues with DHCP client MTU causing interface configure loops when advanced options are present #8507
- Fixed issues with the Hyper-V `hn(4)` driver and ALTQ #8954
- Fixed issues with Hyper-V `hn(4)` interfaces dropping UDP6 traffic when transmit checksums were enabled #9019
- Fixed an issue with IGMP proxy failing to start on PPPoE interfaces #8935
- Fixed an issue with IPv6 Transmit checksums not being disabled when hardware checksums were set to be disabled #8980
- Updated mpd to 5.8_8 to address issues with Orange MTU #8995
- Fixed PPPoE service name checks to allow `:` and other alphanumeric characters #9002
- Fixed PHP errors when creating QinQ entries #9109
- Fixed the MAC address shown when editing a LAGG entry to always show the hardware MAC for each NIC and not the currently active address, which is no longer accurate for LAGG members #8937
- Fixed a PHP error when setting an interface address to act as a DHCP server from the console, when no other DHCP servers are already configured #9144
- Fixed a situation where editing a VLAN interface caused all other VLAN interfaces with the same parent to be reconfigured, which led to several other issues #9115

> **Warning:** Editing a VLAN parent interface can still cause problems. If this becomes an issue on a firewall, consider moving from using the untagged parent to having that traffic be tagged so that the parent interface is not assigned or in use. #9154
>
> Known issues include:
>
> - PPPoE instances on VLANs will not reconnect after the interface is reconfigured #9148
> - VLAN interfaces that use IPv6 tracking may lose their addresses #9136

Hardware/Platform

- Fixed handling of EFI console when a device boots from UEFI, where `vidconsole` is not valid #8978
- Fixed PHP errors in switch configuration on platforms including integrated switches
- Added support for SG-5100 hardware watchdog

Note: Enable the Watchdog daemon under **System > Advanced** on the **Miscellaneous** tab, and then reboot and enable it in the BIOS with a timeout longer than the timeout configured in the GUI.

User Management / Authentication

- Fixed handling of privileges on the **All** group that were previously ignored #9051

> **Warning:** Check the privileges on the **All** group before upgrading to avoid unintended privileges for accounts being respected that were not honored before

- Added GUI options to control `sshguard` sensitivity and whitelisting to allow users to fine-tune the behavior of the brute force login protection #8864
- Added an option to enable SSH agent forwarding (disabled by default) #8590
- Fixed inconsistencies with ssh settings in the configuration #8974
- Fixed PHP errors with ssh settings #8606
- Added support for LDAP client certificates on authentication servers (Factory only) #9007
- Fixed an issue with **Local Database** authentication when using non-English languages in certain cases, such as with Captive Portal #9086

Captive Portal

- Fixed Captive Portal RADIUS NAS Identifier default values to include the zone name #8998
- Restored the ability to set a custom NAS Identifier on Captive Portal RADIUS settings #8998
- Fixed issues with Captive Portal logout popup #9010
- Fixed handling of the login page displayed when RADIUS MAC Authentication fails #9032
- Fixed username sent in RADIUS accounting with MAC-based authentication #9131
- Fixed an issue with the blocked MAC address redirect URL #9114

WebGUI / Dashboard

- Fixed `nginx` restart handling when toggling GUI web server options under **System > Advanced**, **Admin Access** tab
- Fixed empty crash reports after upgrade #8915
- Added CDATA protection to common name fields so they can safely contain international characters #9006

Firewall Rules / Aliases / NAT

- The `filterdns` daemon has been rewritten, solving a number of issues with the old implementation, including:
  - Fixes `filterdns` triggering every 16 seconds even when DNS records have not changed #7143
  - Fixes invalid FQDN entries in aliases causing an alias table to fail silently #8001
  - Fixes `filterdns` failing on a regular basis #8758
- Fixed `/etc/rc.kill_states` not correctly parsing `pfctl` output #8554
- Fixed formatting of alias names to still wrap but not replace underscores #8893

- Fixed PHP errors from `filter_rules_sort()` when a configuration contains no rules #8993
- Fixed PHP errors when creating schedules #9009
- Fixed PHP errors when creating entries on NAT pages #9080
- Fixed PHP errors from `easyrule` when no aliases are present #9119
- Fixed "Drag to reorder" description in rule list when rule drag-and-drop is disabled #9128

Traffic Shaping (ALTQ/Limiters)

- Fixed issues with Limiter queue display on upgraded configurations #8956
- Fixed the default limiter scheduler to match previous version (WF2Q+) #8973
- Added scheduler information to the limiter information page #8973

Packages

- Fixed issues with package installation causing problems when crossing major PHP versions #8938
- Fixed PHP errors when installing packages #9067

Backup/Restore

- Added schedule (cron) support to AutoConfigBackup #8947
- Fixed issues with AutoConfigBackup restoring a configuration from a different host #8901
- Fixed the AutoConfigBackup menu from the deprecated package still showing when the package is no longer present #8959
- Fixed an issue with **Reinstall Packages** hanging when run from **Diagnostics > Backup & Restore** #8933
- Fixed issues with multiple `<rrddata>` tags in `config.xml` #8994
- Fixed a race condition in package operations after a configuration restore that could lead to no packages being reinstalled #9045
- Fixed issues with the External Config Locator not finding a `config.xml` in `/config` #9066
- Fixed an issue where packages may not be reinstalled during a configuration restore performed immediately after a fresh install #9071
- Fixed a `stream_select()` error when restoring packages #9102

Wake on LAN

- Fixed issues with ordering of entries in Wake on LAN #8926
- Added top control buttons to Wake on LAN for **Add** and **Wake all Devices** when there are more than 25 entries #8943

NTP

- Fixed issues with NTP status when using `noquery` in the default permissions along with a specific ACL for localhost #7609

Logging / Notifications

- Fixed an issue with log file sizes `>= 2^32/2` #9081
- Fixed PHP errors when saving log settings #9095
- Added a checkbox to disable TLS certificate verification for SMTP notifications #9001

Install/Upgrade

- Added a FAT partition to the installer memstick to make it easier to restore a `config.xml` file during the install process. Also includes a copy of the license and a README. #9104
- Fixed PHP errors in upgrade code for IPsec #9083

Miscellaneous

- Fixed HTTPS proxy authentication support for connections on the firewall itself #9029
- Clarified wording of **Kernel PTI** options on **System > Advanced**, **Miscellaneous** tab #9026
- Added a Save button to **Status > Traffic Graphs** to store default settings to use when loading the page #8976
- Added support for `nvme` controllers to the S.M.A.R.T. diagnostics page #9042

3.3.5 2.4.4 New Features and Changes

Significant Changes

OS Upgrade Base Operating System upgraded to FreeBSD 11.2-RELEASE-p3. As a part of moving to FreeBSD 11.2, new hardware support is included for C3000-based hardware.

PHP 7.2 PHP upgraded to 7.2, which required numerous changes to syntax throughout the source code and packages.

Routed IPsec (VTI) Routed IPsec is now possible using using FreeBSD `if_ipsec(4)` Virtual Tunnel Interfaces (VTI). #8544 (See also: *Routed IPsec (VTI)*)

IPsec Speed Improvements The new **Asynchronous Cryptography** option under the IPsec **Advanced Settings** tab can dramatically improve IPsec performance on multi-core hardware #8772

Default Gateway Group The default gateway may now be configured using a Gateway Group setup for failover (each gateway on a different tier), which replaces Default Gateway Switching. #8187

Limiter AQM/Queue Schedulers Limiters now include support for several Active Queue Management (AQM) methods and Queue Scheduler configurations such as FQ_CODEL. #6620 (See also: pfSense PR #3941)

Certificate Subject Requirements The Certificate Manager and OpenVPN wizard now only require the **Common Name** to be set, and all other fields are optional. #8381

AutoConfigBackup is free! AutoConfigBackup now integrated and free for all to use. (See also: *Using the AutoConfigBackup Service*)

DNS over TLS The DNS Resolver now includes support for DNS over TLS as both a client and a server, including for domain overrides. #8388 #8030 #8431

Captive Portal Authentication Captive Portal authentication is now integrated with the User Manager system. Captive Portal instances may now use RADIUS, LDAP, or Local Authentication like other integrated services. The firewall will migrate existing Captive Portal RADIUS settings to the User Manager automatically on upgrade.

Captive Portal HTML Design and Usability The default Captive Portal page has been redesigned. Controls have also been added which allow for the logo and background images and Terms of Service text to be customized without editing and uploading custom HTML code. #8793

Integrated Switch Improvements Netgate devices with integrated switches such as the SG-3100 and XG-7100 can now configure per-port speed and duplex settings, discrete port configuration interfaces can now be tied to switch ports for up/down status, and LAGG support is also now available (Load Balance mode only)

Security

- FreeBSD SA for CVE-2018-6922: Resource exhaustion in TCP reassembly FreeBSD-SA-18:08.tcp
- FreeBSD SA for CVE-2018-3620, CVE-2018-3646: L1 Terminal Fault (L1TF) Kernel Information Disclosure FreeBSD-SA-18:09.l1tf
- FreeBSD SA for CVE-2018-6923: Resource exhaustion in IP fragment reassembly FreeBSD-SA-18:10.ip
- FreeBSD SA for CVE-2018-14526: Unauthenticated EAPOL-Key Decryption Vulnerability FreeBSD-SA-18:11.hostapd
- FreeBSD SA for CVE-2018-6924: Improper ELF header parsing FreeBSD-SA-18:12.elf
- FreeBSD errata notice for LazyFPU remediation causing potential data corruption FreeBSD-EN-18:08.lazyfpu
- Fixed a potential XSS vulnerability via GUI rule separators pfSense-SA-18_06.webgui #8654
- Fixed a potential XSS via custom GUI/dashboard settings pfSense-SA-18_07.webgui #8726
- Fixed a potential authenticated ACE vulnerability pfSense-SA-18_08.webgui #8843
- Upgraded strongSwan to 5.6.3 to address a buffer underflow leading to denial of service (CVE-2018-5388) #8746
- Updated default cryptographic settings for OpenVPN, IPsec, and Certificates #8594
- Changed the included DH groups to those defined in RFC 7919 #8582
- Added stronger IPsec Pre-Shared Key usage warnings, and a button to generate a secure PSK #8667
- Disabled OpenVPN compression by default on new instances for security reasons due to VORACLE – Users should strongly consider disabling compression on OpenVPN instances if they pass unencrypted data such as HTTP to arbitrary Internet sites #8788
- Patched OpenSSH for CVE-2018-15473, username enumeration/disclosure through malformed packets.
- Changed from `sshlockout_pf` to `sshguard` for monitoring failed logins and locking out offenders, this allows the lockout to work on IPv4 and IPv6 and also terminates states when adding offenders to the block list #7694 #7695

Errata

Warning: Third party packages from **alternate repositories** are causing problems for users with the upgrade process and also with post-upgrade behavior. These packages have never been supported, and had to be manually added by users outside of the GUI.

Due to the major changes required for FreeBSD 11.2 and PHP 7.2, third party packages from alternate repositories cannot be present during the upgrade. There is no way to predict if a third party package supports the new version or will cause the upgrade itself to fail.

The upgrade process will automatically remove `pfSense-pkg-*` packages installed from alternate repositories. After the upgrade completes, the user can reinstall these packages. Packages from **alternate repositories** will not appear in the **Installed Packages** list in the GUI, and must be entirely managed in the command line.

This change does not affect packages installed from the official pfSense® package repository.

- Removed options for the deprecated FEC LAGG Protocol #8734

Certificates

- Changed the Certificate Manager and OpenVPN wizard to only require the **Common Name** for the CA/Cert subject #8381
- Updated default cryptographic settings Certificates #8594
- Added support for OCSP Must-Staple certificates in the GUI (and ACME package) #8418
- Changed CRL support from using an abandoned PHP OpenSSL module patch to a pure PHP implementation compatible with PHP 7.2 #8762
- Fixed issues with several areas not properly parsing CA fields properly when they were not in the expected order #8801
- Changed the default CA and Certificate create action from "Import…" to "Create an internal…" #8851

DNS

- Added DNS over TLS for upstream forwarders to the DNS Resolver #8388
- Added DNS over TLS server support to the DNS Resolver #8030
- Added DNS over TLS options for DNS Resolver Domain Override #8431
- Fixed editing DNS Resolver ACLs in non-English languages #8539
- Added a DNS Resolver status page #8430
- Clarified that "Register DHCP leases in the DNS Resolver" only works for IPv4 addresses #8592
- Added IPv6 representation of IPv4 addresses in DNS Resolver DNS Rebinding checks #8750
- Fixed disabling the DHCP Server on interfaces when the DNS Resolver **DHCP Registration** option is enabled (Only one enabled interface is required) #8120
- Added advanced option for `qname-minimization` to the DNS Resolver #8028
- Fixed an issue with IDs when editing or deleting DNS Forwarder host override entries #8767

Dynamic DNS

- Added Dynamic DNS client for DigitalOcean DNS #8478
- Fixed Dynamic DNS clients usage of custom check IP services #8664
- Added Dynamic DNS client for Azure #7769
- Updated DNSimple Dynamic DNS client to use DNSimple API v2 #8071
- Fixed handling of username and password fields for custom Dynamic DNS entries #8782

Routing/Gateways

- Added the ability to set a Gateway Group as the default gateway. #3781 #8187
- Extended the maximum Gateway monitoring **Probe Interval** #8593
- Fixed handling of Gateway Group **Trigger Level** #8586
- Fixed inconsistency in display and usage of units for Gateway latency #8477
- Upgraded FRR to 5.0.1 for compatibility with FreeBSD 11.2 #8449
- Fixed FRR BGP MD5 support #8407
- Fixed handling of Router Advertisement preferences #6237

IPsec

- Added routed IPsec using FreeBSD `if_ipsec(4)` VTI #8544
- Added a GUI option to the IPsec **Advanced Settings** tab for Asynchronous Cryptography which can dramatically improve IPsec crypto operation performance on multi-core hardware #8772
- Added IPsec identifiers to **Status > IPsec** #8598
- Fixed a JavaScript variable issue in IPsec IKE Phase 1 causing the Key Length field to be blank in some browsers such as IE #8543
- Added IPsec mobile client options to configure different (virtual) IP addresses per user #8292
- Added IPsec mobile client options to configure different DNS servers per user #8644
- Updated default cryptographic settings for IPsec #8594
- Changed the default behavior of an IPsec Phase 1 to rekey as needed #8540
- Fixed handling of per-user IPsec rules from an authentication server #8765
- Added warnings and hints to IPsec encryption and hash choices about potentially insecure selections #8766
- Fixed an issue with handling IP Alias VIPs with CARP parent after an interface up/down event #8768

OpenVPN

- Disabled compression by default for new OpenVPN client and server instances for security reasons #8788
- Changed OpenVPN Authentication to use an asynchronous authentication plugin which avoids stalling server traffic during the authentication process, especially noticeable on down/broken authentication servers #7905
- Fixed display of **Bridge Route Gateway** options on OpenVPN tap bridge servers #8658
- Fixed handling of LDAP fields in the OpenVPN wizard and brought the options in line with current LDAP server options #8605
- Updated default cryptographic settings for OpenVPN #8594
- Added missing OpenVPN compression options (`stub-v2` and plain `compress`) #8788

DHCP Server

- Fixed validation of custom DHCP options #8534
- Fixed a situation where DHCPv6 was configured for LAN when the LAN interface was not assigned #8048
- Fixed an issue with XMLRPC synchronization of DHCP static mappings #8721

Interfaces / VIPs

- Removed IPv4 and IPv6 settings from the Interface configuration for assigned OpenVPN/GIF/GRE/Routed IPsec instances, since the IP addresses are managed by the parent config not interfaces.php #8687
- Fixed an HTTP_REFERER issue when changing the LAN IP address in the Setup Wizard #8524
- Fixed an HTTP_REFERER issue when changing an interface IP address while accessing the GUI from the same interface #8822
- Fixed handling of the FreeBSD 11.2-BETA dhclient MTU value #8507
- Added PPPoE multi-link over single link to allow users with a supported provider to have a larger MTU #8737
- Fixed a PPPoE MTU issue with ORANGE FR #8595
- Fixed QinQ interface assignment #8446
- Fixed radvd/IPv6 when using a LAN bridge #8429
- Fixed deleting IP Alias VIPs outside an interface subnet where a gateway exists in the same subnet #4438
- Fixed handling of IP Alias and CARP VIP subnet mask/prefix autodetection #8741
- Fixed a panic in IPv6 fragment logging #8499
- Fixed handling of DHCP option 77 in the DHCP client #7425
- Fixed deleting Interface Group members which are disabled #8800
- Fixed MAC address spoofing for bridge interfaces #8138
- Fixed an issue with string termination when creating interfaces through the pfSense PHP module #8683
- Fixed an issue where changing a LAGG could cause a VLAN using that LAGG as a parent interface to lose its association with the LAGG #8527

Integrated Switches

- Added GUI controls to configure LAGG on integrated switch ports (Load Balance mode only)
- Added GUI controls to configure Speed/Duplex for switch ports on integrated switches
- Added the ability to tie the status of an assigned VLAN interface to a switch port for integrated switches
- Added Switch Status to status.php for platforms with a switch #8525
- Fixed an issue switching between Port VLAN and 802.1q VLAN mode on integrated switches #8422
- Fixed an SNMP error on hardware with integrated switches #8600
- Added **Preserve Switch Configuration** option when restoring `config.xml` to keep the current active switch settings instead of those from the imported configuration to help with hardware transitions

Hardware/Platform

- Added support for the new SG-5100
- Fixed an issue with ARM hardware not completely halting when shut down (SG-3100 and SG-1000)
- Fixed HDMI hotplug issues on Minnowboard Turbot hardware (MBT-2220 and MBT-4220)
- Fixed SG-1000 autonegotiation for 10baseT speed and duplex #7532

User Management / Authentication

- Added a visible warning to the user when default password has not been changed #8596
- Fixed configuration descriptions user management operations and added logging #8548
- Fixed escaping of LDAP search parameters #8626
- Fixed an OS issue with adding a group to a user when creating the user #8553
- Fixed handling of LDAP bind credentials #8583
- Removed some legacy code from `auth.inc` #8742
- Fixed Group selections after an input error in the User Manager #8622
- Fixed inconsistent usage of `sshdkeyonly` in `system_advanced_admin.php` #8403
- Added SSH configuration option to require **both** Key **and** Username+Password authentication at the same time #8402
- Replaced `radius.inc` by pear-Auth_RADIUS #7024
- Fixed synchronization of User Manager group scope and operating system groups #7013
- Fixed logging and display of GUI user authentication source IP address when the user logs in through a proxy #8813
- Fixed logging and display of GUI user authentication sources to show what source authorized the login (e.g. LDAP, RADIUS, Local, Fallback) #8816

Captive Portal

- Integrated Captive Portal authentication into the User Manager to enable support for LDAP #5112
- Updated Captive Portal HTML/CSS to a modern design and added controls to customize images and ToS without uploading custom HTML #8793
- Fixed deleting **Allowed Hostnames** and **Allowed IP Addresses** entries in Captive Portal when a zone is disabled #8530
- Added support for setting Captive Portal traffic quotas #8202
- Added display of a custom username when Captive Portal is set to *None* for the authentication type #8361
- Changed handling of Called-Station-Id/Calling-Station ID to send a MAC address instead of IP address when using RADIUS authentication #4294
- Changed to a standardized NAS-Identifier when using RADIUS authentication #3686
- Corrected accounting updates not being sent when expected #8655
- Fixed an issue with XMLRPC synchronization of Captive Portal settings #8806

WebGUI / Dashboard

- Enabled HTTP2 for the Web GUI server #8552
- Updated the text and links in the HTML footer #8733
- Fixed display of available swap with multiple swap disks in the **System Information** Dashboard widget #8587
- Updated text in the Setup Wizard #8753
- Moved the simplepie RSS reader code to a FreeBSD port for easier updates #6998
- Fixed handling of the **Inverse** option in the Traffic Graphs Dashboard Widget #8367
- Fixed issues with the GUI following upgrade progress #8519
- Added a line to display the current GUI user viewing the Dashboard in the System Information Widget #8817

Firewall Rules / NAT / Shaping

- Added CoDel, FQ-CoDel, PIE and FQ-PIE AQMs to limiters #6620
- Fixed firewall ruleset errors related to VIPs and outbound rules #8518 #8408
- Added validation for IPv6 NPt input #8575
- Fixed a race condition in NAT reflection filter rules that could lead to a ruleset load failure #8604
- Fixed viewing the list of Port Forwards when a user only has the "WebCfg - Firewall: NAT: Port Forward" privilege #8563
- Fixed an issue with default field selection when editing Firewall Rules #8597
- Added code to prevent nested alias loops #8101
- Added interface groups support for NAT rules #1933
- Fixed a case where invalid IPv6 NAT rules could be generated #8437
- Fixed a case where IPv6 Neighbor Discovery and other similar valid messages sent from the unspecified address (`::`) were not allowed by default #8791

- Added **Select All** functionality to firewall and NAT rules #8812
- Fixed IPv6 address form field format tooltip #8834

Packages

- Fixed situation where the firewall would get stuck attempting to reinstall packages after restoring a configuration when there is no Internet connection #7604
- Added a new tag for package services, `<starts_on_sync/>`, to allow packages to declare that they start themselves during the sync process, which lets packages opt out of a (second) forced start at boot and during interface events #8850

  See also: #8620

Miscellaneous

- Fixed display of stored Load Balancer custom settings #8704
- Fixed handling of `loader.conf` and `loader.conf.local` so it will not remove customized options that override defaults #8571
- Fixed the restoration process for a `config.xml` from USB during install to remove RRD data so that the data does not indefinitely stay in `config.xml` #7634
- Fixed handling of special characters in L2TP user passwords #7623
- Fixed handling of sample bounds with custom timer periods on **Status > Monitoring** #6477
- Changed the crash reporter so that users can download the reports locally rather than submitting to a server #8764
- Added more redacted XML tags to status.php #8819
- Changed status.php to use `ifconfig -va` to show more detail, including attached SFP devices with certain network interface drivers #8860

3.3.6 2.4.3-p1 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- FreeBSD SA for CVE-2018-8897 FreeBSD-SA-18:06.debugreg
- FreeBSD EN for CVE-2018-6920 and CVE-2018-6921 FreeBSD-EN-18:05.mem
- Fixed a potential LFI in pkg_mgr_install.php pfSense-SA-18_04.webgui #8485
- Fixed a potential XSS in pkg_mgr_install.php pfSense-SA-18_05.webgui #8486

Misc

- Added a check to avoid creating route-to rules for proxy ARP addresses
- Corrected alias name input validation text referring to well-known and registered ports #8409
- Corrected the list of pf reserved keywords to prevent aliases from using invalid custom names #8445
- Fixed an issue with Captive Portal access rules being left behind on disconnect #8441
- Fixed an issue with pressing Enter in the filter field of diag_pftop.php #8494
- Fixed an issue with invalid rules generated due to the presence of IPv6 Alias VIPs #8408
- Fixed an issue with IPsec mobile Pre-Shared Keys and iOS devices #8426
- Fixed an issue with selecting a gateway when switching a firewall rule away from IPv4+IPv6 mode #8447
- Fixed firewall rules generated by the OpenVPN wizard #8391
- Fixed handling of OpenVPN RADIUS attribute firewall rules #8480
- Fixed handling of XMLRPC user/group synchronization when that section is disabled on the primary #8450
- Fixed input validation to allow named services to be used in firewall rules rather than numbers alone #8410
- Fixed issues with IP alias VIPs on Localhost at boot time #8393
- Increased the default Firewall Maximum Table Entries value to 400000 to cope with the increased size of the IPv6 bogon address lists #8417
- Updated SimplePie RSS to 1.5.1 #8423
- Added more fields to the list that status.php uses to redact private information #8394

3.3.7 2.3.5-p2 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- FreeBSD SA for CVE-2018-8897 FreeBSD-SA-18:06.debugreg
- FreeBSD EN for CVE-2018-6920 and CVE-2018-6921 FreeBSD-EN-18:05.mem
- Fixed a potential XSS vector in RRD error output encoding #8269 pfSense-SA-18_01.packages
- Fixed a potential XSS vector in diag_system_activity.php output encoding #8300 pfSense-SA-18_02.webgui
- Fixed a potential LFI in pkg_mgr_install.php #8485 pfSense-SA-18_04.webgui
- Fixed a potential XSS in pkg_mgr_install.php #8486 pfSense-SA-18_05.webgui
- Changed sshd to use delayed compression #8245
- Added encoding for firewall schedule range descriptions #8259

Misc

- Added an option to disable HSTS for the GUI web server #6650
- Added filtering to pfTop page
- Added ospf6d to the routing log
- Change get_interface_subnet() to use configured value if available
- Corrected sethelp call on firewall_rules_edit.php #8242
- Fixed an issue with selecting a gateway when switching a firewall rule away from IPv4+IPv6 mode #8447
- Fixed an issue with the address family selection for remote syslog servers using IPv6 #8323
- Fixed a problem when IPsec bypasslan was enabled while the LAN interface is disabled or doesn't have an IP address #8239
- Fixed config.xml corruption handling
- Fixed input validation for Certificate SAN values to disallow IP addresses for FQDN/Hostname entries #8275
- Fixed issues with OpenVPN when using a /31 IPv4 Tunnel Network #8261
- Fixed NTP Status server time for zones with minute offsets (fractions of an hour) #8129
- Fixed selection of IPv6 gateways when creating a new firewall rule #8053
- Fixed various pf "busy" errors when the ruleset is reloaded
- Improved handling of aliases that mix IP addresses and FQDNs #8290
- Improved update repository controls
- Increased the default Firewall Maximum Table Entries value to 400000 to cope with the increased size of the IPv6 bogon address lists #8417

3.3.8 2.4.3 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- FreeBSD-SA-18:01.ipsec
- Kernel PTI mitigations for Meltdown (optional tunable) FreeBSD-SA-18:03.speculative_execution.asc
- IBRS mitigation for Spectre V2 (requires updated CPU microcode) FreeBSD-SA-18:03.speculative_execution.asc
- Added a CPU Microcode update mechanism (cpuctl module, sysutils/devcpu-data port)
- Imported a FreeBSD patch to fix boot issues when running as a hypervisor guest on AMD Family 15h processors (FreeBSD PR #213155)
- Added validation for RRD parameters to ensure passed filenames are valid #8269
- Fixed a potential XSS vector in RRD error output encoding #8269 pfSense-SA-18_01.packages
- Fixed a potential XSS vector in diag_system_activity.php output encoding #8300 pfSense-SA-18_02.webgui
- Fixed a potential XSS vector in traffic_graphs.widget.php settings #8302 pfSense-SA-18_03.webgui
- Fixed a potential CSRF issue in service control request processing #8296

- Enabled CSRF protection for all dashboard widgets #8301
- Added encoding for firewall schedule range descriptions #8259
- Changed sshd to use delayed compression #8245
- Increased PHP-FPM resources on systems with over 1GB RAM to improve performance #8125
- Imported a netstat fix for ARM platforms to improve performance and reduce CPU usage, especially on the Dashboard #8237
- Fixed a memory leak in the pfSense_getall_interface_addresses() function in the pfSense PHP module #8249
- Hardware support for the XG-7100, including:
  - C3000 NIC support (factory installations only)
  - C3000 SoC support (factory installations only)
  - Marvell 88E6190 switch support (factory installations only)

Traffic Shaping / Limiters

- Fixed hangs due to Limiters and pfsync in HA #4310
- Added the Chelsio *cxl* driver to the list of ALTQ capable interfaces #7607
- Fixed an issue with limiters that had fractional bandwidth values #8091
- Changed status_queues.php to provide 'realtime' statistics #8185

IPsec

- Changed IPsec Phase 1 to allow selecting both IPv4 and IPv6 so the local side can allow inbound connections to either address family #6886
- Changed IPsec Phase 1 to allow configuration of multiple IKE encryption algorithms, key lengths, hashes, and DH groups #8186
- Fixed a problem when IPsec bypasslan was enabled while the LAN interface is disabled or doesn't have an IP address #8239
- Added IPv6 LAN Network to the IPsec LAN bypass list #8321

OpenVPN

- Fixed an error message encountered by a few users when manually killing OpenVPN connections #8266
- Added an OpenVPN tap bridge configuration option to push the bridged interface address to clients as a route-gateway for routes/redirects #8267
- Added an option to the DNS Resolver which allows registering the CN of OpenVPN clients as hostnames #6847
- Added an option to OpenVPN clients and servers to suppress creation of IPv4 or IPv6 gateway addresses for an interface #6848
- Fixed issues with OpenVPN when using a /31 IPv4 Tunnel Network #8261
- Updated the OpenVPN wizard with the current UDP and TCP protocol selections #8298
- Added the interface for a VPN to the OpenVPN client and server list screens

Notifications

- Changed SMTP notifications handling so they are batched, to avoid sending multiple e-mail messages in a short amount of time #4031
- Added a notification when the firewall boot sequence is complete #7643

Dashboard

- Fixed issues with the IPsec dashboard widget causes GUI failure #6318
- Changed the Dynamic DNS Widget so it shows the description of custom entries to identify them #7843
- Fixed a reference to deprecated updateGatewayDisplays() function in the Gateways dashboard widget #8303
- Added a setting to the temperature widget to display readings in Fahrenheit 8205
- Changed the picture widget so the picture is stored on the firewall filesystem and not in config.xml to reduce the size of backup data #8371
  - On upgrade, pictures will be moved out of config.xml, so backup this file separately if it is important

DHCP

- Added an option to the DHCP Server Dynamic DNS configuration to set the server key algorithm #6621
- Added DDNS Client Updates option to DHCPv4 #7131
- Fixed handling of the DHCPv6 DDNS reverse zone key #6319
- Fixed DHCPv4 static mappings so that multiple MAC for same DHCP address or hostname are allowed #8220
- Fixed a potential issue in detecting primary/secondary node in a failover configuration
- Improved DHCP relay destination interface discovery
- Fixed DHCPv6 lease display for entries that were not parsed properly from the lease database #7413

Dynamic DNS

- Added an option for RFC 2136 Dynamic DNS server key algorithm #8244
- Added an option for RFC 2136 source address used to send updates #8278
- Fixed issues with Dynamic DNS updates using a gateway group when the primary route is down #8333
- Added GoDaddy Dynamic DNS provider

Interfaces / VIPs

- Fixed issues on assign_interfaces.php with large numbers of interfaces #6400
- Fixed handling of CARP VIPs on disabled interfaces at boot time #6677
- Fixed issues with radvd being enabled on a disconnected interface #6974
- Fixed issues with rtsold on VLAN interfaces #7412
- Fixed issues with dhcp6c lock files after unclean shutdown when using “Do not wait for an RA” on IPv6 WAN interface #8106

- Added a feature to allow pppoe on a CARP VIP so it will only be active on whichever node is master #8184
- Fixed an error when editing PPP interfaces on a system with no VIPs #8322
- Added VLAN priority tagging for DHCPv6 client requests #8200
- Added support for configuring the DUID type for an IPv6 interfaces #8191
- Allow custom INIT string for PPP modem SIM Pin and APN settings
- Added an indicator for disabled interfaces on status_interfaces.php
- Fixed an issue with the PPP linkup and linkdown scripts and cellular modems
- Fixed an issue where the combination of CARP with bridging could lead to a deadlock #8056

Packages

- Fixed reinstall process for missing packages #8183

Captive Portal

- Fixed Pass-through MAC automatic additions so it does not add duplicate entries #8226
- Fixed a missing global definition in Captive Portal pass-through MAC removal #8238
- Fixed Captive Portal voucher sync errors when vouchers are expired or disconnected while the secondary node is master #8317
- Fixed Captive Portal voucher synchronization between HA nodes #7972

Certificates

- Fixed automatic SAN handling when the CN of a certificate contains a space #8252
- Fixed input validation for Certificate SAN values to disallow IP addresses for FQDN/Hostname entries #8275

Gateways/Routing

- Fixed handling of the Router Lifetime value on services_router_advertisements.php so it allows a value of *0* #7502
- Added ospf6d to the routing log
- Allow recursive aliases to be used with static routes

Rules/NAT

- Fixed various pf "busy" errors when the ruleset is reloaded
- Fixed issues with editing firewall rules in non-English languages that contain single quotes in translated strings #8219
- Added an option to disable drag-and-drop of firewall and NAT rules
- Added a check to prevent 1:1 NAT rules with missing information from being added to the ruleset
- Added firewall rule tracking ID to rule list (in counter tooltip) and firewall rule edit page #8348

- Fixed cases where automatic or scripted rules were not getting tracking IDs #8353
- Added a check to prevent automatic outbound firewall rules with missing information from being added to the ruleset #8360

Users/Authentication

- Fixed issues with XMLRPC user account synchronization causing GUI inaccessibility on secondary HA nodes #7469
- Fixed an issue where a user with no privileges could not logout #8297
- Increased maximum username length from 16 to 32 characters to catch up to the current allowed length in FreeBSD
- Fixed required field markings on LDAP authentication server configuration fields #8337
- Fixed display of the LDAP host when testing the GUI authentication source #8338

Misc

- Fixed NTP Status server time for zones with minute offsets (fractions of an hour) #8129
- Added support for custom shutdown scripts in /usr/local/etc/rc.d #8182
- Fixed a references to an undefined function while restoring a config.xml file from an older version #8231
- Added support to diag_packet_capture.php to capture traffic on the loopback interface #8257
- Fixed an issue with the RAM disk warning pop-up appearing when no changes were made #8268
- Fixed an issue with the address family selection for remote syslog servers using IPv6 #8323
- Silenced warnings from sysctl that otherwise went to stderr
- Added a disk size check to ZFS to prevent it from being used on disk which are too small to contain the OS and swap space #7308
- Added a check to prevent pfSense-upgrade from running as a non-root user #7762
- Added an option to disable the IGMP Proxy service #8356
- Fixed an issue with package handling when restoring a configuration that contains a branch configuration that is not valid for the target system version #8208

3.3.9 2.4.2-p1 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- Updated OpenSSL to address CVE-2017-3737 and CVE-2017-3738 FreeBSD-SA-17:12.openssl
- Fixed a potential authenticated command execution issue in certificate data handling #8153 pfSense-SA-17_10.packages.asc
- Fixed a potential XSS issue in status_filter_reload.php #8143 pfSense-SA-17_11.packages.asc

Misc

- Fixed an issue with the subnet mask not being preserved properly when editing existing 1:1 NAT entries #8112
- Fixed an indexing issue when deleting Host Override entries from the DNS Forwarder #8159
- Fixed logging for L2TP and PPPoE server login/logout events #8164
- Removed ix from the ALTQ interface list since ALTQ support for the ix driver is not currently viable #7378
- Fixed a premature session timeout issue on pages which update exclusively using AJAX, such as status_graph.php #8116
- Fixed ping_hosts.sh so it does not unnecessarily run a CARP check when there are no IPsec hosts to ping #8172
- Fixed a missing global variable declaration in interface IP address detection
- Fixed issues with local authentication when using translated languages

3.3.10 2.4.2 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- Updated to OpenSSL 1.0.2m to address CVE-2017-3736 and CVE-2017-3735
- FreeBSD-SA-17:10.kldstat
- FreeBSD-SA-17:08.ptrace
- Fixed a potential XSS vector in status_monitoring.php #8037 pfSense-SA-17_07.packages.asc
- Fixed a potential XSS vector in diag_dns.php #7999 pfSense-SA-17_08.webgui.asc
- Fixed a potential XSS vector on index.php via widget sequence parameters #8000 pfSense-SA-17_09.webgui.asc
- Fixed a potential XSS in the widgetkey parameter of multi-instance dashboard widgets #7998 pfSense-SA-17_09.webgui.asc
- Fixed a potential clickjacking issue in the CSRF error page

Interfaces

- Fixed PPP interfaces with a VLAN parent when using the new VLAN names #7981
- Fixed issues with QinQ interfaces failing to show as active #7942
- Fixed a panic/crash when disabling a LAGG interface #7940
- Fixed issues with LAGG interfaces losing their MAC address #7928
- Fixed a crash in radvd on SG-3100 (ARM) #8022
- Fixed an issue with UDP packet drops on SG-1000 #7426
- Added an interface to manage the built-in switch on the SG-3100
- Trimmed more characters off the interface description to avoid console menu output line wrapping on a VGA console
- Fixed handling of the VIP uniqueid parameter when changing VIP types
- Fixed PPP link parameter field display when a VLAN parent interface was selected #8098

Operating System

- Fixed issues resulting from having a manually configured filesystem layout with a separate /usr slice #8065
- Fixed issues updating ZFS systems created ZFS using an MBR partition scheme (empty /boot due to bootpool not being imported) #8063
- Fixed issues with BGP sessions utilizing MD5 TCP signatures in routing daemon packages #7969
- Updated dpinger to 3.0
- Enhanced the update repository selection choices and methods
- Updated the system tunables that tell the OS not harvest data from interrupts, point-to-point interfaces and Ethernet devices to reflect the new name/format for FreeBSD 11
- Changed ruleset processing so that it retries if another process is in the middle of an update, rather than presenting an error to the user
- Fixed some UEFI boot issues on various platforms

Certificates

- Fixed invalid entries in /etc/ssl/openssl.cnf (only affected non-standard usage of openssl in the cli/shell) #8059
- Fixed LDAP authentication when the server uses a globally trusted root CA (new CA selection for "Global Root CA List") #8044
- Fixed issues creating a certificate with a wildcard CN/SAN #7994
- Added validation to the Certificate Manager to prevent importing a non-certificate authority certificate into the CA tab #7885

IPsec

- Fixed a problem using IPsec CA certificates when the subject contains multiple RDNs of the same type #7929
- Fixed an issue with enabling IPsec mobile client support in translated languages #8043
- Fixed issues with IPsec status display/output, including multiple entries (one disconnected, one connected) #8003
- Fixed display of multiple connected mobile IPsec clients #7856
- Fixed display of child SA entries #7856

OpenVPN

- Added an option for OpenVPN servers to utilize "redirect-gateway ipv6" to act as the default gateway for connecting VPN clients with IPv6, similar to "redirect-gateway def1" for IPv4. #8082
- Fixed the OpenVPN Client Certificate Revocation List option #8088

Traffic Shaping

- Fixed an error when configuring a limiter over 2Gb/s (new max is 4Gb/s) #7979
- Fixed issues with bridge network interfaces not supporting ALTQ #7936
- Fixed issues with vtnet network interfaces not supporting ALTQ #7594
- Fixed an issue with Status > Queues failing to display statistics for VLAN interfaces #8007
- Fixed an issue with traffic shaping queues not allowing the total of all child queues to be 100% #7786
- Fixed an issue with limiters given invalid fractional/non-integer values from limiter entries or passed to Captive Portal from RADIUS #8097

Rules/NAT

- Fixed selection of IPv6 gateways when creating a new firewall rule #8053
- Fixed errors on the Port Forward configuration page resulting from stale/non-pfSense cookie/query data #8039
- Fixed setting VLAN Priority via firewall rules #7973

XMLRPC

- Fixed a problem with XMLRPC synchronization when the synchronization user has a password containing spaces #8032
- Fixed XMLRPC Issues with Captive Portal vouchers #8079

WebGUI

- Added an option to disable HSTS for the GUI web server #6650
- Changed the GUI web service to block direct download of .inc files #8005
- Fixed sorting of Services on the dashboard widget and Services Status page #8069
- Fixed an input issue where static IPv6 entries allowed invalid input for address fields #8024
- Fixed a JavaScript syntax error in traffic graphs when invalid data is encountered (e.g. user was logged out or session cleared) #7990
- Fixed sampling errors in Traffic Graphs #7966
- Fixed a JavaScript error on Status > Monitoring #7961
- Fixed a display issue with empty tables on Internet Explorer 11 #7978
- Changed configuration processing to use an exception rather than die() when it detects a corrupted configuration
- Added filtering to the pfTop page
- Added a means for packages to display a modal to the user (e.g. reboot required before package can be used)

Dashboard

- Fixed display of available updates on the Installed Packages Dashboard widget #8035
- Fixed a font issue in the Support Dashboard widget #7980
- Fixed formatting of disk slices/partitions in the System Information Dashboard widget
- Fixed an issue with the Pictures widget when there is no valid picture saved #7896

Packages

- Fixed display of packages which have been removed from the repository in the Package Manager #7946
- Fixed an issue displaying locally installed packages when the remote package repository is unavailable #7917

Misc

- Fixed interface binding in ntpd so it does not erroneously listen on all interfaces #8046
- Fixed a problem where restarting the syslogd service would make sshlockout_pf process orphans #7984
- Added support for the ClouDNS dynamic DNS provider #7823
- Fixed an issue in the User and Group Manager pages when operating on entries immediately after deleting an entry #7733
- Changed the setup wizard so it skips interface configuration when run on an AWS EC2 Instance #6459
- Fixed an IGMP Proxy issue with All-multicast mode on SG-1000 #7710

3.3.11 2.4.1 New Features and Changes

Security / Errata

- Fixes for the set of WPA2 Key Reinstallation Attack issues commonly known as KRACK #7951
- Changed upgrade handling to use the pkg-static binary to prevent errors when moving to new major FreeBSD version
- Fixed a VT console race condition panic at boot on VMware platforms (especially ESXi 6.5.0U1) #7925
- Fixed a bsnmpd problem that causes it to use all available CPU and RAM with the hostres module in cases where disk drives are present without media inserted #6882
- Fixed an upgrade problem due to FreeBSD 11 removing legacy ada aliases, which caused some older installs to fail when mounting root post-upgrade #7937
- Changed the boot-time fsck process the ensure the disk is mounted read-only before running fsck in the preen mode.

Known Issues

- The VLAN changes mentioned in the **Interfaces** section may prevent PPP sessions from working on VLANs in some cases, see #7981

Interfaces

- Changed the VLAN interface names to use the 'dotted' format of FreeBSD, which is shorter and helps to keep the interface name smaller than the limit (16) This fixes the 4 digit VLAN issues when the NIC name is 6 bytes long.
- Improved the 'Assign Interfaces' console process to automatically stop when there are no more interfaces to assign
- Improved the 'Set interface IP address' console process to accept 'IP/mask' notation
- Fixed wireless client interfaces so they do not reconfigure wireless on a link up event, or else they can get stuck in a loop #7960
- Fixed setting VLAN Priority in VLAN interface configuration #7748

Dashboard

- Fixed a problem with the Picture Dashboard widget when it does not have a picture defined #7896
- Fixed time display for UTC in the NTP Dashboard Widget #7714
- Fixed an IPsec widget error when it would get back null data after a session ended #6318
- Improved error checking to prevent dashboard widget parsing errors

DNS

- Added an option for the DNS Resolver (Unbound) to serve expired records from the cache after their TTL expires which can improve speed in some cases #7814
- Fixed the DNS Resolver (Unbound) to allow snoop from localhost by default, otherwise "dig +trace" or "drill -T" queries from the firewall itself fail #7884

XMLRPC

- Fixed XMLRPC Sync to prevent a lock that would never be unlocked
- Fixed XMLRPC sync to ensure a proper empty array is returned instead of NULL, so that the last item of a section can be removed without error #7953

Misc

- Fixed Captive Portal voucher test and expire pages #7939
- Added UEFI 32 and UEFI 64 filenames defined inside a pool to dhcpd.conf #7949
- Fixed operation of the "Reset All States on WAN IP Change" GUI setting #7921
- Changed OpenVPN to retry client auth when it fails by default (auth-retry nointeract) #7506
- Changed the Cryptographic Accelerator module options to allow both the AES-NI and Crypto modules to be loaded at the same time #7810
- Added URL fingerprinting to the login page CSS
- Added the device serial/id to the console and SSH menu banner #7968
- Fixed "Unknown Step Values" in certain RRD graph cases #6860

3.3.12 2.4 New Features and Changes

Operating System / Architecture changes

- Upgrade of base OS to FreeBSD **11.1-RELEASE-p1**
- Added support for Netgate ARM-based systems such as the SG-1000
- **32-bit support has been deprecated and removed** – There are no images available for 32-bit (x86/i386) Intel architecture systems
- **NanoBSD has been deprecated and removed** – There are no images available for NanoBSD, use a full install instead
- Started using the FreeBSD installer instead of the old style installer (installation procedures have all changed)
  - The installer now supports UEFI #4044
  - If the new installer image will not boot on a specific piece of hardware, see *Troubleshooting Boot Issues*
  - The installer now supports ZFS
  - Added support to the new installer to copy an existing config.xml off an MS DOS formatted USB drive (formerly known as "PFI") #7689

  - Added support to the new installer to optionally recover config.xml off an existing installation drive (works with UFS and ZFS) #7708

- Fixed issues with major version base upgrades via pkg
- Changed cryptodev to load as a kernel module #5976

Security / Errata

- Converted various parts of the GUI to use POST instead of GET when performing actions that change the firewall state (e.g. delete or enable/disable an item) to avoid potential issues with cross-site request forgery and unintentional repeating of actions #4083
- FreeBSD 11.1 includes MAP_GUARD protection to protect against attacks such as Stack Clash
- pfSense-SA-17_07.packages
- A number of base system packages have been updated to address security issues, including dnsmasq, perl, cURL, and others.

Firmware Branch Behavior / Upgrading From Snapshots

- To control how a firewall obtains updates, visit **System > Update**, **Update Settings** tab:
- For users currently running 2.3.x-RELEASE:
  - *Stable*, which is the default behavior, will upgrade the firewall to 2.4-RELEASE when it is complete, but will not upgrade to 2.4-RC
  - *Development Snapshots* will upgrade the firewall to 2.3.5 development snapshots
  - *Next major version* will upgrade the firewall to 2.4-RC
- For users tracking pfSense® software version 2.4 snapshots:
  - *Stable*, which is the default behavior, will upgrade the firewall to 2.4-RC and eventually 2.4-RELEASE
  - *Development Snapshots* will cause the firewall to continue tracking snapshots, bypassing 2.4-RELEASE and continuing on to 2.4.1 development images

Known Issues

- Some systems may not be able to boot 2.4 installation images, for example, due to UEFI compatibility changes. These are primarily BIOS issues and not issues with the installer images. Upgrading from 2.3.x should still work on affected hardware.
- Users with ESXi or VMware Workstation may experience a boot-time crash during hardware detection, due to a race condition in the FreeBSD VT console code. This crash is infrequent and does not affect most users or most boot attempts, but since it is a race condition it can manifest randomly. To avoid the crash, configure the VM to use the `syscons` console rather than `vt` by editing `/boot/loader.conf.local` and adding this line:

```
kern.vty=sc
```

Cleanup

- Misc code cleanup, removal of patches that were no longer necessary or were inefficient
- Replaced multiple local copies of PHP PEAR libraries with updated copies using their official sources #3734
  - Notably, local static copies were replaced by their FreeBSD ports counterparts: pear, pear-XML_RPC2, pear-Net_IPv6, pear-Crypt_CHAP, pear-Mail, pear-Net_Growl
  - Code that relied on the old files was updated to use the current or replaced versions
- Removed all references to GLXSB (it was 32-bit only) #6755
- Removed all code in the builder and pfSense for handling the NanoBSD platform
- Removed all calls to conf_mount_rw / conf_mount_ro, since they were only required for NanoBSD
- Improved help text in various parts of the GUI

Wireless

- FreeBSD 11 contains an updated 802.11 stack with numerous improvements
- Wireless interfaces **must** be created on the **Wireless** tab under **Interfaces > Assignments** before they can be assigned! #6770

Firewall / Rules / NAT / Aliases

- Fixed issues with synproxy rules on a WAN/LAN style bridge #6769
- Fixed issues with limiters on rules that utilize NAT #4326
- Fixed issues with limiters used in conjunction with a transparent proxy or other local redirect rule #7050
- Fixed expansion of "Other" type VIP subnet entries in NAT destination drop-down selections #6094
- Fixed NAT rules so that when a port forward is disabled, its associated firewall rule is also disabled #6472
- Fixed handling of "URL Table (IPs)" and "URL (IPs)" when the file is hosted a server using HTTPS with a self-signed certificate #4766
- Show firewall rule descriptions in a column when viewing the log on new installs, upgrades retain their existing setting #7323
- Fixed firewall states showing a negative value for total bytes processed #7075
- Fixed handling of Port Forwards so they do not make up new destination information when a configured against a DHCP interface that does not currently have an address
- Fixed VLAN Priority pf syntax #7744
- Fixed a problem where pf scrub did not properly re-fragment unusual but valid IPv6 fragments, resulting in overlapping fragments #7485
- Fixed confirmation prompt handling when deleting a firewall state from diag_dump_states.php #7827
- Changed display of 1:1 NAT rules to match other firewall pages #7728

Traffic Shaping

- Added extra warnings to traffic shaping pages when the firewall has no interfaces capable of using ALTQ shaping #7032
- Fixed handling removal of shaping rules when deleting an interface #7231
- Added upgrade code to work around broken shaper rules from older wizard code #7434
- Fixed the Traffic Shaper so it shows interface names for disabled interfaces, rather than an 'empty' placeholder.
- Fixed handling of the priority field for different ALTQ shaper types

OpenVPN

- Upgraded OpenVPN to 2.4.x. #7054
  - This is a significant upgrade which includes support for a wide variety of new features, including AEAD ciphers such as AES-GCM.
  - AES-GCM can be accelerated by AES-NI, and is supported in SSL/TLS modes (not shared key) #7068
  - Added support for TLS Encryption as an optional TLS Key usage type. This encrypts the control channel, providing privacy and protocol obfuscation #7071
  - Added ECDH options to OpenVPN server and client options ("ECDH Only" choice for DH, ECDH Curve selection) #7063
  - Restructured the compression options to include LZ4 support and the new "compress" directive which replaces "comp-lzo" which has been deprecated. The old options remain for now, but are labeled "Legacy" #7064
  - Changed protocol selection for OpenVPN clients and servers because OpenVPN 2.4 treats "udp" and "tcp" as dual stack now #7062
    - Added "multihome" option in relevant protocol cases so OpenVPN will reply back using the address used to receive a packet #7062
  - Changed the DNS Server fields in the OpenVPN server options so they can define either IPv4 or IPv6 DNS servers to push to clients #7061
  - Added IPv6 support to status_openvpn.php and the OpenVPN widget #2766
  - Removed uses of the deprecated "tun-ipv6" OpenVPN directive, OpenVPN now always assumes IPv6 is enabled #7054
  - Replaced uses of the deprecated "client-cert-not-required" directive with its functional replacement "verify-client-cert none" #7073
  - Added support for Negotiable Crypto Parameters (NCP) to control automatic cipher selection between clients and servers #7072
  - NOTE: OpenVPN 2.4 handles CRL verification differently than previous versions, passing through validation to the library rather than handling it internally. This can cause some certificates to fail validation that may have passed previously. In particular, if a certificate is removed from a CRL, it may still fail validation until all copies of the CRL have been rewritten.
- Improved the help text on OpenVPN Client-Specific Overrides #7053
- Fixed issues with OpenVPN clients on dynamic or tunneled IPv6 interfaces (e.g. GIF) #6663
- Added locking to prevent issues with OpenVPN instance startup #6132
- Check OpenVPN server/client option visibility changes per mode #7331 #7451

- Added an OpenVPN GUI option for “fast-io” to clients and servers #7507
- Added an OpenVPN GUI Option for “sndbuf” and “rcvbuf”, using the same value for both #7507
- Removed references to the defunct OpenVPN client manager port #7568
- Removed references to unused “Address Pool” setting in OpenVPN #7567
- Fixed OpenVPN server port validation to disallow “0”, while still allowing it for a client port, which is the same meaning as blank/empty #7565
- Fixed OpenVPN help text for route_no_exec #7575
- Fixed description of the address assignment behavior for Tunnel Network fields in OpenVPN clients and servers #7573
- Remove the GUI option for “resolv-retry infinite” from OpenVPN, it is always enabled #7572
- Fixed the OpenVPN wizard so it better handles a user choosing a different type of authentication server than a previous run of the wizard #7569
- Fixed OpenVPN Auth Digest Algorithm selection so it does not use duplicate/alias names in the list, and added upgrade code to fix existing entries on upgrade so they use the actual digest name and not an alias #7685
- Fixed show/hide behavior of fields on vpn_openvpn_client.php in chrome #7451
- Changed OpenVPN wizard certificate input validation and encoding so it matches the standards of the current certificate manager #7854
- Fixed the OpenVPN wizard so it creates an OpenVPN server instance using current proper defaults #7864

IPsec

- Upgraded strongSwan to version 5.6.0
- Changed the default strongSwan logging levels such that IKE SA, IKE Child SA, and Configuration Backend all default to “Diag” #7007
- Added an option to set the Rekey Margin for IPsec tunnels in the Phase 1 settings
- Added RADIUS accounting support for mobile IPsec when accounting is enabled on the Authentication Server entry
- Added checks to prevent simultaneous/repeated calling of vpn_ipsec_configure() by /etc/rc.newipsecdns
- Added DH Groups 22, 23, 24 to IPsec Phase 2 selection for compatibility, but they should not normally be used for security reasons #6967

Certificate Management

- Added a check to ensure that the public key of the Certificate matches its private key when importing Certificate Authority and Certificate entries to prevent mismatching keys from being imported #6953
- Fixed error handling when creating a Certificate from the User Management section, failed actions will no longer fail silently #6953
- Fixed handling of Certificates generated from an imported CA when no starting serial number was set #6952
- Fixed handling of Certificate Authority deletion so that it does not remove associated certificates #6947
- Added “in-use” testing for Certificate Authority entries and disabled the delete action for CAs which are actively in use #6947
- Fixed choosing an existing user certificate when adding a certificate to an existing user #7297

- Added the ability for the certificate manager to sign a CSR using an internal CA #7383
- Added the ability to set the certificate type and SAN attributes in a Certificate Signing Request #7527
- Restructured how certificate types and SANs are handled in the cert manager when making a Cert/CSR/Signing, so each section can properly use the controls #7527 #7677
  - It is now possible to add SANs and EKUs to certificates when signing using the certificate manager
  - NOTE: Attributes such as SANs and KU/EKU cannot be copied from a CSR when signing due to a deficiency in OpenSSL's x509 functions (they do not support "copy_extensions" at this time); These attributes must be specified manually when signing
- Fixed "server" certificate detection to key off of the EKU For "TLS Web Server Authentication" since nsCertType has been deprecated
- Added SAN, KU, and EKU information in an info block for each entry in the the certificate list #7505
- Added the ability to use a wider range of characters in certificate fields as laid out by RFC 4514 #7540
- Added a useful error message when there is no private CA with which to create a new user certificate from within the user manager #7585
- Fixed the User Manager so it adds the username as the first SAN when making a user certificate at the same time a user is created #7666
- Added another possible Certificate Signing Request Armor string when validating on import #7383

Dynamic DNS

- Fixed response parsing for DNSimple Dynamic DNS #6874
- Fixed handling of password in Dynamic DNS entries to allow special characters #6688
- Changed CloudFlare and GratisDNS to use separate hostname and domain entry to handle TLDs with multiple components #6778
- Fixed the Save and Force Update button for RFC2136 Dynamic DNS #7291
- Fixed RFC2136 Dynamnic DNS updates at boot time #7295
- Added the 'local' directive to RFC2136 Dynamic DNS so updates are sourced correctly #7446
- Fixed options text and display for IPv4 DNS and Verify SSL on Dynamic DNS clients #7588
- Fixed issues with Dynamic DNS entries utilizing gateway groups for their interface #7719
- Added DreamHost Dynamic DNS support #7321

DHCP Server / Relay

- Fixed handling of DHCPv6 lease status when there are no leases #6717
- Fixed issues with DHCP Relay not working #6658
- Added input validation to prevent the DHCP server from being configured on interfaces that do not have enough addresses for clients (/31, /32) #6930
- Fixed issues with the DHCP Relay options display getting out of sync with checkbox settings #7155
- Fixed static DHCP lease edits updating BIND zones #3710
- Fixed checks for DHCP Relay when editing additional DHCP pools
- Fixed handling of forced Dynamic DNS hostnames for DHCPv6 static mappings #7324

ARP / NDP

- Fixed static ARP handling when creating or editing DHCP static mappings #6821
- Added error checking for static ARP entries to ensure both an IP address and MAC address are entered, and to ensure that both exist before an entry is applied #6969
- Improved the detail displayed on the ARP table view #6822
- Added an expiration field to the NDP list

Captive Portal

- Adapted Captive Portal to work without multi-instance ipfw patches #6606
- Fixed Captive Portal instances to select "No Authentication" for a zone by default, since it is the default behavior #7591
- Fixed links to the Captive Portal MAC management page so they include the zone name #7591

XMLRPC

- Switched to pear-XML_RPC2 and removed the outdated static client files
- Fixed handling of XMLRPC sync using a username other than "admin" #809

Routing/Gateways

- Removed "route change" patches and updated code that relied on the deprecated behavior #6828
- Fixed handling of default routes when a default gateway is removed or disabled #6659
- Fixed discovery of IPv6 gateway for assigned OpenVPN interfaces #6016
- Fixed issues with a missing default gateway/route on certain PPPoE links after reconnect or IP address change #6495
- Fixed some 'route: writing to routing socket: Invalid argument' warnings during boot time
- Added a log message for gateway events that shows that an alarm was raised/cleared
- Added a check to not run dpinger when an IPv6 address has the tentative flag even after the timeout
- Added a delay to allow dpinger time to properly initialize before using results

Interfaces / Virtual IP Addresses

- Removed Device Polling as it was no longer useful #7021
- Improved stability of the igb(4) driver #7149 #7166
- Fixed handling of rc.newwanipv6 when run from dhcp6c so it only runs when required and not for any change #7145
- Fixed handling of SIGTERM and SIGKILL in dhcp6c #7185
- Fixed dhcp6c not starting until an RA is received #5993
- Fixed a PPP service name error with certain providers, such as T-Mobile #6890
- Fixed 3G service status so it does not report misleading information #4287

- Added support for the IPv6 AUTO_LINKLOCAL flag on bridge interfaces
- Disabled DAD on stf interfaces to fix problems with dpinger
- Added an option to use static IPv6 over an IPv4 PPP parent (e.g. PPPoE) #7598
- Removed unused WINS code for L2TP #7559
- Improved L2TP Server DNS input validation #7560
- Added a test to disable internal L2TP users when activating RADIUS, to follow the behavior stated in the GUI #7561
- Fixed L2TP section log shortcut #7564
- Fixed upgrade handling of wireless interfaces #7809

NTP

- Added support for the ntpd "pool" directive to make better use of servers in NTP pools #5985
- Fixed time display on the NTP widget to show server time #7245
- Added support for NTP to process PGRMF NMEA sentences (Garmin-specific) #7193
- Added an absolute offset statistic to NTP monitoring graph display #7548

User Management / Authentication

- Fixed delays during bootup when LDAP is enabled for user authentication #6367
- Added privileges to control which users can view and/or clear notices #7051
- Added an authentication cache mechanism for GUI authentication from a remote server (e.g. LDAP, RADIUS) so the authentication is checked periodically (default: 30s) instead of on each page load #7097
- Added protocol selection (PAP, MD5-CHAP, MS-CHAPv1 and MS-CHAPv2) to RADIUS authentication server options #7111
- Added the username to the page to display when adding user privileges #7586
- Standardized privilege page and sorting between users and groups #7587
- Added a log message if a user tries to save the configuration but has the 'deny config write' permission
- Added "auth_check" type of simple test that a page can use to verify a user is logged in and has access, using less cpu, which is better for AJAX data polling
- Fixed certificate chain verification issues with LDAP authentication using intermediate CAs #7830
- Fixed PHP errors when STARTTLS fails for LDAP authentication

Packages

- Fixed issues with snort, squid/clamav, and squidGuard when /var is in a RAM disk #6878
- Fixed handling of custom_php_deinstall_command during post-deinstall of a package #7401
- Changed package related calls to get_pkg_info() to use the new pkg metadata mechanism

Console / Menu

- Added options to the console reboot menu selection to reboot into single user mode or run filesystem checks #6639

OS Upgrade

- Fixed issues when upgrading to 2.4 with a stale package .inc that caused a PHP error #6920
- Changed the upgrade script to use reroot instead of reboot for updates that do not include a new Kernel #6045

SNMP

- Added a workaround to prevent the hostres module from being used with bsnmpd on VMware Virtual Machines that have a cd0 device, which caused 100% CPU usage #6882

Services

- Converted all mpd-based features (e.g. PPPoE and L2TP server) to MPD5 if they used an older version #4706
- Removed unused and non-functional SMART service handling and e-mail configuration #6393
- Fixed IGMP Proxy failing to recognize an upstream interface #6099

WebGUI

- Added support for multiple languages, currently that list includes:
  - US English (Default), Bosnian, Chinese (Simplified, China), Chinese (Taiwan), Dutch, German, Norwegian Bokmal, Polish, Portuguese (Brazil), Russian, Spanish, Spanish (Argentina)
- Changed the design of the login page for the WebGUI to a more modern style, with several color choices available
- Added URL fingerprinting to JavaScript and CSS file references to improve client-side behavior when files change between versions #7251
- Updated Logo to the new logo and made it a vectorized SVG image for better scaling
- Updated favicon to the new logo and added multiple sizes for different platforms
- Completed work to mark required fields on GUI pages #7160
- Fixed long hostnames overlapping the “time” title in the monitoring graphs #6138
- Fixed CIDR/Prefix selector handling for IPv4/IPv6 #7625
- Removed the Gold menu
- Fixed handling of info block content inside tables #7504

- Improved handling of PHP errors for user-entered PHP code on diag_command.php
- Fixed alignment of the the IPv6 over IPv4 input fields #7128
- Optimized retrieval of Traffic Graph data to reduce spikes in the graphs and load on the firewall
- Fixed a problem with the traffic graphs not respecting the theme colors #6746
- Revised setup wizard wording and links

Dashboard

- Rewrote Dashboard AJAX updating in a centralized and optimized way to reduce load, improve accuracy, and increase speed
- Added a new Customer Support dashboard widget, enabled by default and on upgrade
- Changed the way AJAX updates are handled on the Dashboard widgets to improve efficiency and fix issues with some widgets refreshing in a timely manner
- Added filters to more dashboard widgets #7122
- Added customization for dashboard widget names
- Fixed Interface Statistics dashboard widget issues with interfaces in a "down" state
- Fixed formatting issues with the Interface Statistics dashboard widget #7501
- Added the ability to place multiple copies of widgets on the dashboard, optional for each widget
- Added a line to display detected CPU cryptographic hardware, such as AES-NI, in the System Information dashboard widget even if the module isn't loaded #7529
- Fixed CPU package/core count displayed on the System Information dashboard widget
- Changed how pkg metadata is handled to reduce the load on the Dashboard and reduce unnecessary calls to the pkg server for the System Information dashboard widget update check, and for the Installed Packages dashboard widget
- Changed CPU usage calculation in the System Information dashboard widget to avoid sleep() in an AJAX call
- Fixed the IPsec widget tunnel status to handle newer strongSwan childid format #7499
- Fixed error when saving Wake on LAN dashboard widget without any WoL entries
- Fixed a problem where traffic could be counted twice in traffic graphs #7751
- Fixed a problem with the Installed Packages dashboard widget when no packages are installed #7811
- Changed date formats of some fields on the Dashboard to be more consistent #7805
- Added an option to the Interface Statistics dashboard widget to rotate the table (put interfaces in rows instead of columns) to improve the display on firewalls with numerous interfaces #7501

pftop

- Removed the "size" option from pftop as it had no effect, use the "bytes" option instead #7579
- Removed the 'peak' and 'rate' views for pftop since they only work in interactive mode with cached data, not batch mode which is used by the WebGUI #7580
- Fixed path to an old copy of the pftop WebGUI page in obsolete list #7581

DNS

- Changed /etc/hosts such that the FQDN is listed first, except for localhost, so that dnsmasq will properly reverse resolve hostnames #7771
- Fixed a problem where the DNS Search Domain List was not being populated into radvd.conf #7081
- Enabled Python support for Unbound #7549
- Added a control to disable automatically added host entries in Unbound
- Changed the way unbound is started at boot time on systems with DHCP6 WANs

Misc

- Added hardware support and detection for new Netgate models
- Changed the User Agent passed to outbound requests from pfSense to include more accurate host information
- Added the User Agent to the request data when updating the Bogons list
- Fixed growl and SMTP notifications so performing a test saves first, so the new settings are used as expected #7577
- Fixed loading issues with PHP extensions #6628
- Removed symbolic links for configuration files that redirected items from /etc/ to /var/etc/ #5538
- Added the ability to filter Packet Captures by MAC address #6743
- Updated status.php with new info and changed its output organization #7047
- Fixed a problem where a proxy defined for use by the firewall could not use HTTPS when using proxy authentication #6949
- Improved RAM disk backups and file management #7098
- Changed the way RAM disk contents are handled when enabled #5897
- Changed various support functions to better facilitate translation to additional languages
- Fixed interface name display on the Router Advertisement configuration page #7133
- Fixed various issues with handling of unusually formatted, but valid, IPv6 addresses #7147
- Improved error handling when a client is logged when it attempts to poll data via rrd_fetch_json.php #6748
- Fixed various issues when the configuration backup count was set to 0 (disabled) #7273
- Fixed handling of "0" for the number of backups to retain in the configuration history #7273
- Fixed an issue with long configuration change descriptions leading to wrapping issues in certain cases such as AutoConfigBackup #6363

- Fixed an issue with installing packages from a backup when restoring using the External Configuration Locater on the first boot post-install #7914

3.3.13 2.3.5-p1 New Features and Changes

New features and changes for this release of pfSense® software:

Security / Errata

- Updated OpenSSL to address CVE-2017-3737 and CVE-2017-3738 FreeBSD-SA-17:12.openssl
- Fixed a potential authenticated command execution issue in certificate data handling #8153 pfSense-SA-17_10.packages.asc
- Fixed a potential clickjacking issue in the CSRF error page
- Fixed a potential XSS issue in status_filter_reload.php #8143 pfSense-SA-17_11.packages.asc

Misc

- Fixed an issue in the User and Group Manager pages when operating on entries immediately after deleting an entry #7733
- Fixed sorting of Services on the dashboard widget and Services Status page #8069
- Fixed display of available updates on the Installed Packages Dashboard widget #8035
- Fixed display of packages which have been removed from the repository in the Package Manager #7946
- Fixed the OpenVPN Client Certificate Revocation List option #8088
- Fixed an issue with the Pictures widget when there is no valid picture saved #7896
- Fixed an indexing issue when deleting Host Override entries from the DNS Forwarder #8159
- Fixed a premature session timeout issue on pages which update exclusively using AJAX, such as status_graph.php #8116
- Fixed ping_hosts.sh so it does not unnecessarily run a CARP check when there are no IPsec hosts to ping #8172
- Fixed a missing global variable declaration in interface IP address detection

3.3.14 2.3.5 New Features and Changes

The pfSense® software version 2.3.x release is a Security and Errata maintenance release. 2.4.x is the primary stable supported branch. If the firewall hardware is capable of running 2.4.x, consider upgrading to that release instead.

Updating to 2.3.5 from 2.3.4 on an amd64 installation that could otherwise use 2.4.x requires configuring the firewall to stay on 2.3.x as follows:

- Navigate to **System > Update**, **Update Settings** tab
- Set **Branch** to **Security / Errata Only**
- Navigate back to the **Update** tab to see the latest 2.3.x update

If the update system offers an upgrade to 2.3.5 but the upgrade will not proceed, ensure the firewall has correct versions of the repository configuration and upgrade script for 2.3.x by running the following commands from the console or shell:

```
pkg install -fy pfSense-repo pfSense-upgrade
```

Firewalls running 32-bit (i386) installations of pfSense software do not need to take any special actions to remain on 2.3.x as they are unable to run later versions.

Operating System / Architecture changes

- Upgrade of base OS to FreeBSD **10.3-RELEASE-p20**
- Fixed issues with major version base upgrades via pkg

Security / Errata

- pfSense-SA-17_07.packages
- Fixes for the set of WPA2 Key Reinstallation Attack issues commonly known as KRACK in wpa_supplicant and hostapd (FreeBSD-SA-17:07.wpa)
- A number of base system packages have been updated to address security issues, including dnsmasq, perl, cURL, and others.

Interfaces

- Added support for the IPv6 AUTO_LINKLOCAL flag on bridge interfaces
- Added an option to use static IPv6 over an IPv4 PPP parent (e.g. PPPoE) #7598
- Added IPv6 Prefix Delegation interface selection
- Improved input validation for GIF interfaces #7789

Dashboard

- Rewrote Dashboard AJAX updating in a centralized and optimized way to reduce load, improve accuracy, and increase speed
- Added a new Customer Support dashboard widget, enabled by default and on upgrade
- Changed the way AJAX updates are handled on the Dashboard widgets to improve efficiency and fix issues with some widgets refreshing in a timely manner
- Changed how pkg metadata is handled to reduce the load on the Dashboard and reduce unnecessary calls to the pkg server for the System Information dashboard widget update check, and for the Installed Packages dashboard widget
- Improved error checking to prevent dashboard widget parsing errors
- Fixed a variable conflict in the NTP Status Dashboard widget #7795
- Fixed a problem with the Picture Dashboard widget when it does not have a picture defined #7896
- Changed IPsec Dashboard Widget tunnel status to handle newer strongSwan childid format #7499
- Fixed time display for UTC in the NTP Dashboard Widget #7714

WebGUI

- Changed the design of the login page for the WebGUI to a more modern style, with several color choices available
- Added URL fingerprinting to JavaScript and CSS file references to improve client-side behavior when files change between versions #7251
- Updated Logo to the new logo and made it a vectorized SVG image for better scaling
- Updated favicon to the new logo and added multiple sizes for different platforms
- Added an option for sorting the Interfaces menu by description
- Added "auth_check" type of simple test that a page can use to verify a user is logged in and has access, using less cpu, which is better for AJAX data polling
- Improved handling of PHP errors for user-entered PHP code on diag_command.php
- Changed Interfaces menu "(Assign)" to "Assignments" and added support for menu divider bars
- Fixed automatic selection of '128' as prefix/mask for IPv6 address fields #7625
- Replaced Math.trunc with Math.floor to make IE properly handle traffic graphs #7804
- Changed nginx configuration so it does not allow direct download of .inc files #8005
- Fixed hostname input handling on diag_dns.php

Gateways

- Added a delay to allow dpinger time to properly initialize before using results
- Added a log message when gateway alarms are raised/cleared to show the parameters that triggered the alarm
- Reset All States on WAN IP Change option #1629

Rules/NAT/Shaper

- Fixed handling of Port Forwards so they do not make up new destination information when a configured against a DHCP interface that does not currently have an address
- Fixed ALTQ Traffic Shaper PRIQ priority number validation

IPsec

- Added an option to set the Rekey Margin for IPsec tunnels in the Phase 1 settings
- Added RADIUS accounting support for mobile IPsec when accounting is enabled on the Authentication Server entry
- Added checks to prevent simultaneous/repeated calling of vpn_ipsec_configure() by /etc/rc.newipsecdns

Misc

- Fixed an issue with installing packages from a backup when restoring using the External Configuration Locater on the first boot post-install #7914
- Fixed handling of forced Dynamic DNS hostnames for DHCPv6 static mappings #7324
- Fixed several issues with cron job updating and removal
- Added the device serial/id to the console and SSH menu banner #7968
- Changed /etc/hosts such that the FQDN is listed first, except for localhost, so that dnsmasq will properly reverse resolve hostnames #7771

3.3.15 2.3.4-p1 New Features and Changes

The pfSense® software version 2.3.4-p1 errata release is a minor release after *2.3.4* and contains beneficial security and bug fixes.

Security / Errata

- pfSense Security Advisories
  - pfSense-SA-17_05.webgui:
    - Fixed a potential XSS issue in the diag_edit.php file browser #7650
    - Fixed a potential XSS in handling of the 'type' parameter on diag_table.php #7652
    - Fixed validation and a potential XSS in interface names on firewall_nat_edit.php #7651
  - pfSense-SA-17_06.webgui:
    - Added a warning screen to the GUI and prevent access if the client IP address is currently in the lockout table, and also remove the client's connection states #7693

Bug Fixes

Captive Portal

- Fixed Captive Portal RADIUS Authentication to only cache credentials when required to perform reauthentication #7528
- Restored the captive portal feature to view the captive portal page directly from the portal web server as an additional button #7646

Dynamic DNS

- Fixed issues with wildcard CNAME records disappearing from Loopia when doing a DNS update
- Fixed issues with CloudFlare Dynamic DNS
- Fixed Hover Dynamic DNS updates so they Verify the SSL Peer

Logging

- Added syslogd service definition to enable status display and control #4382
- Fixed issues with syslogd stopping when installing or uninstalling some packages #7256

Virtual IP Addresses

- Fixed issues with CARP status display overmatching some VIP numbers #7638
- Fixed pid file handling for choparp (Proxy ARP Daemon)
- Added the ability to sort the Virtual IP address list

DNS

- Fixed diag_dns.php so it will not create an empty alias if name does not resolve
- Fixed diag_dns.php to not show Add Alias if the user does not have privileges to add an alais
- Fixed diag_dns.php to change the update alias button text after adding an alias
- Fixed diag_dns.php to disable the Add Alias button when the host field is changed
- Fixed calls to unbound-control to have the full configuration path specified so they do not fail #7667
- Fixed handling of "redirect" zone entries in the DNS Resolver so they do not produce invalid zones #7690
- Changed the way the DNS Resolver code writes out host entries, so the zones are more well-formed #7690
- Changed the way the DNS Resolver process (unbound) is stopped, to allow it to exit cleanly. #7326

Interfaces

- Fixed DHCPv6 to request a prefix delegation even if no interfaces are set to track6 #4544
- Updated handling of original MAC address retention for interfaces with spoofed MACs
- Fixed an array handling problem when working with gateway entries on the Interface configuration page #7659
- Fixed handling of MSS clamping values for PPPoE/L2TP/PPTP WANs #7675

DHCP

- Fixed an issue where some DHCP Lease information was encoded twice with htmlentities/htmlspecialchars
- Fixed an issue where in some edge cases, a variable was not properly set in a loop, leading to a previous value being reused

Misc

- Removed "/usr/local/share/examples" from obsolete files list, some packages rely on the files being there
- Added a few more items to status.php for support purposes, such as a download button, socket buffer info, and the netgate ID
- Fixed status.php to redact BGP MD5 password/key in output #7642
- Fixed OpenVPN to use is_numeric() to make sure $prefix is not 0
- Changed the "Rule Information" section so it is consistent between firewall and NAT rule pages
- Fixed APU2 detection for devices running coreboot v4.x
- Fixed the tunable description for net.inet.ip.random_id #6087
- Fixed some outdated links for help and support
- Fixed some issues with empty config tags in packages #7624
- Fixed issues with entry IDs after deleting Authentication Server instances #7682

3.3.16 2.3.4 New Features and Changes

Security / Errata

- Updated base OS to FreeBSD 10.3-RELEASE-p19
- FreeBSD/ports Security Advisories
  - Updated ntpd to 4.2.8p10_2 (FreeBSD-SA-17:03.ntp.asc)
  - Updated cURL to 7.54.0 (CVE-2017-7407, CVE-2017-7468)
  - Updated libevent to 2.1.8 (CVE-2016-10197, CVE-2016-10196, CVE-2016-10195)
- pfSense® Software Advisories
  - Fixed encoding of displayed values from DHCP leases to prevent a badly formatted DHCP lease hostname from causing a potential XSS #7497 (pfSense-SA-17_04.webgui)
- See the Certificates section below for an important note about GUI certificate errors on Chrome 58 and later

Certificates

- Improved certificate generation to always include the CN as the first Subject Alternative Name (SAN), which fixes issues with Chrome 58+ #7496
  - To work around an error with the firewall GUI certificate on Chrome 58+, take one of the following actions:
    - Generate and activate a new GUI certificate automatically, from the console/shell: pfSsh.php playback generateguicert
    - Utilize the *ACME package* to generate a trusted certificate for the GUI via Let's Encrypt
    - Create a own new CA/Server certificate and use that for the GUI
    - Activate the local browser "EnableCommonNameFallbackForLocalAnchors" option in Chrome 58 (this setting will be removed by Chrome eventually, so this is only a temporary fix)
- Fixed linking of a certificates to its CA after submitting the signed version of a CSR #7512

Firewall Rules/NAT/Shaper

- Fixed restarting the Load Balancer (relayd) clearing system tables/aliases #7396
- Fixed ruleset generation to notify when an unresolvable alias is encountered by the parser #7421
- Fixed handling of a rule using an empty port alias #7428
- Fixed the traffic shaping wizard handling of SMB rules in Raise/Lower Other Protocols, it was producing an invalid rule #7434
- Fixed handling of alias renaming after input validation #7473
- Fixed handling of long rule descriptions #7294

Dashboard

- Improved formatting in the gateways widget by reducing the numeric precision of displayed values #6841
- Fixed the NTP widget to show the server time instead of client time #7245
- Added a "None" option to Widgets with filtering options #7318
- Added PPPoE uptime display on the Interfaces dashboard widget #6032
- Added filters to more dashboard widgets #7122
- Added BIOS information to the System Information widget
- Added Netgate Unique ID to the System Information widget
  - This identifier for support services is only displayed on the Dashboard for information purposes and is **not** transmitted anywhere automatically by default. In the future, customers can use this identifier when requesting support information from our staff or systems.

Configuration

- Fixed issues restoring a configuration containing packages when the firewall does not have Internet connectivity #6594
- Fixed factory reset when Captive Portal has Vouchers enabled #7508
- Cleaned up unused code in diag_backup.php

Interfaces

- Changed interface handling so it retains the original vendor MAC address at power up when spoofing, so it can be restored without a reboot #7011
- Fixed interface assignment of QinQ interfaces #4669
- Fixed errors in PPP service provider selection when a country without providers is selected #7399
- Fixed input handling when editing static IP address fields on interfaces #7493
- Added the ability for DHCP Client WANs to specify a list of IP addresses from which to reject leases #7510

User Manager / Authentication

- Added a warning to system_authservers.php to warn that RADIUS does not work with IPv6 #4154
- Added a status icon to the User Manager to show if a user is enabled or disabled #7517

General GUI

- Added navigation links to breadcrumbs #7099
- Improved service name support and error handling in pfSenseHelpers.js #7445

DHCP

- Changed dhcpleases so it does not start when DHCP Relay is enabled #6750
- Fixed checks for DHCP Relay being enabled/disabled so they are skipped when editing an additional pool

ARP / NDP

- Added the ability to delete NDP entries #7513
- Added expiration field to NDP listing #7514

Misc

- Fixed DNS issues when upgrading NanoBSD #7345
- Fixed the Reset Demotion Status for CARP to function when the demotion value is negative #7424
- Fixed editing of Host Overrides in the DNS Resolver/Forwarder pages #7435
- Fixed service handling (start/stop/restart) for Captive Portal #7444
- Fixed display of the ALTQ "queue" view in pfTop due to recent changes in the pfTop port #7461
- Added support for the Dynamic DNS Client Hover #7511
- Fixed UTF-8 handling in Base64 decoding on diag_edit.php
- Fixed handling of traffic graph data irregularities #7515
- Added visual separation to the legend on the installed packages list #7203
- Changed SMTP and Growl notification test to use the new, unsaved settings #7516

3.3.17 2.3.3-p1 New Features and Changes

The pfSense® software version 2.3.3-p1 errata release is a minor release after *2.3.3* and contains beneficial security and bug fixes.

Security / Errata

- Updated to FreeBSD 10.3-RELEASE-p17
  - FreeBSD-SA-17:02.openssl (CVE-2016-7055, CVE-2017-3731, CVE-2017-3732)
- Upgraded cURL to 7.53.0 (CVE-2017-2629)

Bug Fixes

- Fixed issues with the upgrade check seeing the version of pfSense-upgrade instead of pfSense in some circumstances. #7343
- Fixed handling of domain-only (@ record) updates for CloudFlare Dynamic DNS #7357
- Fixed a problem with the Dynamic DNS Widget where RFC2136 entries showed an incorrect status #7290
- Fixed Dynamic DNS status widget formatting for medium with browser window #7301
- Fixed a problem with HTML tags showing in certificate description drop-down lists in the Certificate Manager #7296
- Fixed an error loading some older rules with ICMP types #7299
- Fixed display of selected ICMP types for old rules without an ipprotocol option set #7300
- Fixed Log widget filter interface selection with custom interface descriptions #7306
- Fixed the widget Filter All button so it does not affect all widgets #7317
- Fixed the password reset script so it resets the expiration date for the admin account when run, to avoid the user still being locked out #7354
- Fixed the password reset script so it properly handles the case when the admin account has been removed from config.xml #7354
- Fixed input validation of TCP State Timeout on firewall rules so it is not arbitrarily limited to a maximum of 3600 seconds #7356
- Fixed console settings for XG-1540/XG-1541 to use the correct default console #7358
- Fixed initial setup handling of VLAN interfaces when they were assigned at the console before running the Setup Wizard #7364
- Fixed display of OpenSSL and input errors when working in the Certificate Manager #7370
- Fixed Captive Portal “disconnect all” button
- Fixed pkg handling timeouts #6594
- Updated blog URL in the RSS widget
- Removed whirlpool from the list of CA/certificate digest algorithms since it does not work #7370

3.3.18 2.3.3 New Features and Changes

Security / Errata

- Updated to FreeBSD 10.3-RELEASE-p16
  - FreeBSD Security Advisories
    - FreeBSD-SA-16:29.bspatch
    - FreeBSD-SA-16:31.libarchive
    - FreeBSD-SA-16:33.openssh
    - FreeBSD-SA-16:35.openssl
    - FreeBSD-SA-16:37.libc
    - FreeBSD-SA-16:38.bhyve
    - FreeBSD-SA-16:39.ntp
    - FreeBSD-SA-17:01.openssh
  - FreeBSD Errata Notices
    - FreeBSD-EN-16:17.vm
    - FreeBSD-EN-16:18.loader
- pfSense® Software Advisories
  - pfSense-SA-17_01.webgui
    - Fixed validation and encoding on Captive Portal status pages #7019
  - pfSense-SA-17_02.webgui
    - Fixed update_config_field() in wizard.php so it does not pass user input through eval() #7230
  - pfSense-SA-17_03.webgui
    - Added encoding for 'from' and 'to' before output on pkg_mgr_install.php #7225
    - Added encoding for the contents of pkg_filter before output #7227
    - Converted easyrule.php to use a confirmation landing page so that the parameters can be submitted via POST #7228
- Updated numerous third-party libraries and supporting programs
- Changed behavior of fsck during bootup to improve filesystem stability #6340
- Added protection to /etc/ttys to prevent corruption or missing lines

Known Issues

- The Captive Portal **Disconnect All Users** button does not fully disconnect all users PR#3565
- RFC 2136 Dynamic DNS Entries will show red on the Dashboard widget even when correctly updated #7290
- If an OpenVPN server set for *SSL/TLS+User Auth* contains a single user certificate shared between multiple users with different usernames, the **Duplicate Connections** option must be enabled on the server. In this situation, each user must have their own unique certificate or the certificate requirement should be removed (User Auth only). As this configuration is not valid nor a recommended practice, this issue is not considered a bug.

When this condition is present only a single user can connect, additional users may see a client log entry such as "CreateIpForwardEntry: The object already exists".

- Firewall rules without an IP protocol set in the configuration which also have an ICMP type set may not load or display correctly. #7299 #7300

General Info

- Added Packages: tinc, cellular, LCDproc, TFTP Server
- Fixed numerous typos and wording issues
- Added marking for required fields on various pages #7083
- Input validation fixes on various pages
- Cleaned up some unneeded files/pages/functions
- Fixed broken/outdated links

OpenVPN

- Changed OpenVPN RADIUS authentication to send proper NAS-Port-Type, NAS-Port, and NAS-Identifier values #6609
- Added compression option to handle connecting to OpenVPN peers which do not have LZO compiled into their OpenVPN executable #6739
- Added a workaround to block outside DNS on Windows 10 OpenVPN clients to prevent DNS leaks #6719
- Improved OpenVPN server handling when using CARP VIPs in Gateway Groups
- Improved handling of chained/intermediate CAs in OpenVPN #2800
- Changed OpenVPN widget so it updates dynamically #6723
- Adapted the encryption cipher list to the new output format in OpenVPN 2.3.12, also now displays key and block lengths #6849
- Changed OpenVPN server list to display more information
- Improved error message to explicitly state allowable characters for certificate fields in the OpenVPN wizard #6432
- Fixed handling of OpenVPN authentication when the backend server name contains special characters (e.g. '&') #7002
- Fixed saving an OpenVPN instance on a DHCP interface that does not currently have an IP address #7031
- Added an IPv6 Tunnel Network field to OpenVPN Client-Specific Overrides #7053
- Fixed changing between tun and tap mode for OpenVPN Clients
- Changed OpenVPN startup to avoid overwriting its configuration, and to wait for its PID file to be written
- Fixed OpenVPN binding to an IP Alias VIP #7136
- Fixed display of disabled OpenVPN clients #7180
- Fixed handling of "redirect-gateway" in Client-Specific Overrides #6633

IPsec

- Clarified IPsec Key Exchange Version drop-down to specify IKEv1/IKEv2 #6898
- Fixed handling of static routes for IPsec peers on tunnels bound to IP Aliases VIPs with CARP parents
- Fixed MSS clamping for mobile IPsec clients #7005
- Added IPsec to the State Table interface list

Interfaces

- Fixed handling of LAGG MTU when child QinQ interfaces are present #6227
- Improved behavior when using DHCP before RA #5993
- Added the ability to send a DHCP Release from Status > Interfaces, rather than only stopping dhclient
- Fixed issues adding/editing QinQ entries
- Fixed input validation of QinQ entries
- Fixed validation to prevent an interface, interface group, and alias from using the same name #6976
- Updated interface group name validation rules to match limits of the operating system
- Prevented interface group names, interface names, and aliases from starting with `pkg_` to reserve it for packages use (e.g. tinc) #7173
- Added validation to prevent Interface Group Names from containing a dash #7173
- Added validation to prevent Interface Groups from being renamed to an existing name #7183
- Fixed issues with Interface Statistics widget display #7134
- Fixes for interfaces_ppps_edit.php to fix MTU validation, interface friendly names, advanced options expansion
- Changed linkup event handling to ignore events for interfaces that are member of bridges which have no IP address configured
- Fixed input validation for L2TP and PPTP WAN type interfaces #6732
- Added validation to prevent adding duplicate gateways from the Interface configuration page
- Fixed handling of IPv6 checksum options for “Disable hardware checksum offload” #5321
- Fixed handling of the confirmation dialog when deleting a VLAN #6916
- Fixed handling of wireless MAC address spoofing
- Fixed wireless channel changing #6833
- Improved labels and help text for IPv6 tunneling options
- Added the ability for an L2TP or PPTP WAN to use a hostname for the remote gateway #6899

Certificate Management

- Added missing recommended key lengths and digests to certificate manager
- Fixed CRL editing so that certificates already contained the CRL are not displayed

Users / Authentication / Privileges

- Fixed SSH Keyboard-Interactive authentication #6963
- Added STARTTLS to LDAP Authentication Server Configuration
- Improved WebGUI usability when a remote LDAP server is not available
- Fixed issues with local_sync_accounts failing during boot when using an LDAP server on a non-local network or hostname #6857
- Fixed port build options for scponly #7012
- Fixed notifications so that the Mark All as Read button is not shown to users who do not have sufficient privileges to use it #3454
- Added privileges to control display of notices #7051
- Standardized privilege name capitalization
- Fixed issues with low-privilege users accessing Help pages #7139 #7140
- Added a privilege for UPnP & NAT-PMP configuration #7141
- Simplified tcsh prompt and changed the prompt so it respects default terminal colors

Firewall / Rules / NAT / Aliases / States

- Fixed restoring rule type selection after input errors while saving firewall rules
- Fixed a copy/paste error in variable test when validating firewall rule ports.
- Corrected the descriptions and behavior of the Adaptive Start and Adaptive End settings for firewall state handling
- Fixed display of the number of states in the Firewall Rules page
- Moved “Any” to top of protocol list in firewall rules
- Fixed issues with hidden fields on firewall_rules_edit.php #7057
- Fixed issues with moving rules that required scrolling while dragging #6895
- Enhanced ICMP type handling in rules
- Fixed issues when hovering the mouse pointer over aliases on disabled rules making the hint difficult to read #6448
- Fixed handling of firewall rule separators when a NAT associated rule is deleted #6676
- Added field to specify source-hash key for outbound NAT rules
- Fixed issues with **Firewall > NAT > Edit** forgetting destination type selection when input errors occur #6224
- Removed “self” as a destination from NAT 1:1 rules
- Fixed NAT rules so that when a port forward is disabled, its associated firewall rule is also disabled #6472
- Fixed 1:1 NAT address family validation #6927

- Fixed problems with nested aliases containing FQDNs #6982
- Changed the Status > Filter Reload page so it shows the entire filter reload progress, rather than only the last state #6931
- Fixed labels on diag_states_summary.php #6711
- Fixed initial state of confirmation checkboxes on diag_resetstate.php
- Changed Diag > States so it can optionally require a filter before displaying states, to improve handling with large state tables #7069

Traffic Shaping

- Added Chelsio network cards (cxl) to the list of drivers that are capable of using ALTQ #6830
- Fixed the traffic shaper wizard so it uses whole numbers instead of decimals #6779

HA / CARP

- Fixed issues when XMLRPC synchronizes IP Alias type Virtual IP addresses bound to Localhost #7010
- Fixed a bug where the CARP VIP status was incorrect when the interface has more than one CARP VIP

DHCP/DHCPv6 Server / Router Advertisements

- Updated the ISC DHCP Daemon to fix issues with missing hostnames in leases, and removed workarounds that are no longer needed #6840
- Fixed reversed behavior of "Change DHCPv6 display lease time from UTC to local time" #6640
- Fixed incorrect index for edit action on DHCP Leases #7233
- Added an option to force a Dynamic DNS hostname in DHCP/DHCP6 Server settings
- Changed DHCP lease times to always display in 24-hour clock format
- Added an option to allow BOOTP to be specifically disabled in the DHCP Server settings #4351
- Fixed validation to allow URLs for TFTP Server in DHCP Server settings #6634
- Improve dhcpd and dhcpleases reload handling
- Fixed DHCP NTP Server form validation to allow hyphens #6806
- Fixed restore of DHCP6 leases on full install when using MFS /var
- Fixed a problem with the DHCP range being reset if the Setup Wizard was re-run when a custom DHCP range already exists #4820
- Fixed issues with DHCP traffic being blocked with DHCP Relay enabled #6996
- Changed the DHCP/DHCPv6 server GUI so it can be configured (but not run) while DHCP Relay is enabled #6997
- Added Client ID to DHCP Leases display, if present
- Added Client ID to DHCP Mapping list, if present
- Disabled DHCP server on interfaces with subnet >= 31 #6930
- Changed DHCP6 client to allow a prefix size of /59

- Changed DHCP6 server to allow a prefix size of /59 and /61
- Added new "Ignore client identifiers" option to DHCP Server
- Fixed handling of DNS entries for IPv6 static mappings when using delegated prefixes #6768
- Improved the help text for Router Advertisement configuration #6889

DNS / Resolver / Forwarder

- Allow a variable number of DNS servers #5549
- Changed interface boxes in the DNS Resolver so they can be resized
- Fixed sorting of DNS Forwarder hosts and domains in config.xml #6903
- Fixed DNS Resolver (unbound) logging after clearing logs #6915
- Added support for "deny_non_local" and "refuse_non_local" ACLs in the DNS Resolver #6914
- Fixed DNS Server Gateway validation
- Changed behavior of DNS Resolver overrides to only add FQDN entries, not short hostnames #6064
- Fixed issues with DNS Resolver Host Overrides not being updated properly #6712

NTP / GPS

- Fixed display of Prefer/No Select checkboxes invisible when adding entries in NTP Server settings #6788
- Fixed handling of NTP IPv6 restrict clauses
- Fixed setting default NTP access restrictions when there are no custom restrictions #6454
- Fixed NTP status widget IPv6 address handling so addresses are not truncated #4815
- Fixed the NTP Orphan Mode stratum field #7034
- Fixed issues with NTP GPS status
- Fixed a case that could result in an empty 'restrict' line in the NTP configuration #7110
- Added a limit for NTP time source fields so they cannot exceed the maximum number saved to configuration #7164
- Fixed display and behavior issues with NTP ACLs #6984
- Improved parsing of GPS initialization and output, and add support for more GPS output formats and extended status
- Added an autocorrect tool for checksums on GPS initialization commands #7159

Captive Portal

- Changed Captive Portal MACs page to be sortable #6786
- Fixed handling of Captive Portal user bandwidth set to 0 #6872
- Changed Captive portal to send "Admin Reset" as termination cause when disconnecting a user from the WebGUI
- Added option to Captive Portal to include idle time in total session time
- Fix bandwidth limitation settings in Captive Portal MAC passthrough

- Fixed links to view current Captive Portal page for all interfaces #6391
- Converted Captive Portal active sessions to a sortable table
- Added code to hide the client MAC address column in Captive Portal status when MAC filtering is disabled, rather than displaying an empty column
- Added popup with session details to the Captive Portal active sessions list on the status page
- Added button to disconnect all Captive Portal users
- Worked around race condition between captiveportal_disconnect_all() and captiveportal_prune_old()
- Added locking to avoid race conditions between rc.prunecaptiveportal and captiveportal_disconnect_all()
- Reworked logging and RADIUS accounting when disabling a Captive Portal zone or rebooting
- Increased speed of captiveportal_disconnect_all()

Dynamic DNS

- Added the ability to change the URL queried by Dynamic DNS entries to check the external IP address (Services > Dynamic DNS, Check IP Services tab) #6591
- Added support for All-Inkl Dynamic DNS provider
- Added support for duiadns.net Dynamic DNS provider
- Added support for CloudFlare Proxy to Dynamic DNS
- Added Cloudflare Dynamic DNS IPv6 support #6623
- Fixed status checking on Dynamic DNS (RFC2136), updates were always considered successful even on failure #6357
- Fixed handling of multiple RFC2136 entries #6153
- Fixed links in RFC2136 entries in the Dynamic DNS widget #7126
- Fixed HTTP header processing for Dynamic DNS updates
- Fixed handling of custom IPv6 Dynamic DNS in the widget #6922
- Changed Cloudflare and Gratis plus Dynamic DNS to store passwords in base64
- Updated Route 53 Dynamic DNS to fix several reported issues #3973 #6751 #5054
- Fixed handling of ZoneEdit Dynamic DNS when used with a CARP VIP #6992
- Removed excess loops from the Dynamic DNS Widget

Gateways / Routing

- Added the ability to disable gateway monitoring actions without disabling gateway monitoring #3151
- Changed gateway notifications to notify by email and syslog when a gateway goes up or down
- Improved gateway notification mechanisms
- Fixed handling of deleting or disabling static default gateways so they are properly removed from the routing table #6659
- Fixed L2TP WAN dynamic gateway naming #6980
- Fixed status display for unmonitored gateways

- Fixed static blackhole route handling
- Fixed handling of long hostnames on Diagnostics > Routes #6869
- Corrected behavior of disabled static routes #3560
- Created a PHP Shell playback script to view the gateway status from the shell and status output #7046

Notifications

- Fixed SMTP settings test so it properly displays results
- Fixed validation of secure SMTP Connection Modes (SSL/TLS and STARTTLS are mutually exclusive)
- Removed validation of password mismatches when SMTP or Growl notifications are disabled #7129
- Changed format of file_notice() alerts in webgui for easier reading

Graphs / Monitoring

- Changed traffic graphs to use d3.js (Dashboard and Status > Traffic Graphs)
- Moved export button to heading for Status > Monitoring page
- Moved graph labels so long hostnames do not overlap as easily #6138
- Improved error checking in case JSON isn't returned when building graphs #6748
- Added a missing RRD step value to lookup table #6860
- Added support for multiple views in Status > Monitoring graphs (Adds tab shortcuts to different graph views)
- Added a per-view "Refresh Interval" option to Status > Monitoring graphs
- Fixed fix null acronyms and axis label for queues/queuedrops graph in Status > Monitoring
- Enabled Area and Bar graph types for Status > Monitoring graphs

WebGUI

- Added an option to allow display of the firewall hostname on the login page
- Added filtering to widgets where appropriate
- Standardized PHP memory limit configuration
- Fixed formatting issues with the Installed Packages widget #6601
- Improved Compact-RED theme
- Changed service running/stopped icons
- Fixed issues with JavaScript confirmation prompts missing words (e.g. "Are you sure you wish to?") #6972
- Fixed issues with packages that toggle visibility of advanced options areas #7100
- Removed the crash reporter link from the dashboard when a user does not have crash_reporter page access #7043
- Fixed display of Package installation message #7226
- Fixed "" tag processing in package XML handling
- Fixed inconsistent handling of empty/null configuration settings in config.xml #6893

Logging

- Increased filtering tail limit for logging to ensure enough entries will be displayed #6652
- Added a means for packages to request a syslogd socket inside a chroot environment #4898
- Added BIND logging to proper facility #5524
- Improved handling of the TFTP Proxy/xinetd process when it is disabled, to reduce log messages #6308

Misc

- Updated simplepie (RSS Parsing library) to 1.4.3
- Fixed storing of IPv6 addresses so they are always saved in lower case #6864
- Fixed bsnmpd "printcap" log errors #6838
- Fixed a foreach error when restoring a configuration without packages
- Fixed handling of signal traps in the console menu #6741
- Fixed "Goto line #" action on diag_edit.php so pressing the enter key also activates the function
- Changed the PHP Execute feature of Diagnostics > Command so that it does not generate a crash report from a syntax error #6702
- Added enable link to Status > UPnP & NAT-PMP error message if disabled #6689
- Changed the time zone help text to clarify and warn against the use of the Etc time zones that use POSIX style signs, which are the opposite of what most users expect #7089
- Added validation to prevent duplicate Wake on LAN entries
- Fixed permissions on /var/tmp when /var is a RAM disk #7120
- Added a fallback for get_pkg_info() to use pkg info if there is no local copy of the repository catalog
- Removed spurious output from the PHP Shell executable when running a playback script from a command prompt #7045
- Updated status.php with new info and changed its output organization #7246

3.3.19 2.3.2-p1 New Features and Changes

2.3.2 Update 1

- FreeBSD-SA-16:26.openssl - Multiple vulnerabilities in OpenSSL. The only significant impact on pfSense® software is OCSP for HAproxy and FreeRADIUS.
- Several HyperV-related Errata in FreeBSD 10.3, FreeBSD-EN-16:10 through 16:16. See https://www.freebsd.org/relnotes/10-STABLE/errata/errata.html for details.
- Several built-in packages and libraries have been updated, including:
  - PHP to 5.6.26
  - libidn to 1.33
  - curl to 7.50.3
  - libxml2 to 2.9.4

- The hardware serial number is now displayed in the system information widget, or a host UUID if a serial number is not found. This is for display purposes and facilitates users seeking support in identifying their hardware.
- Added encoding to the 'zone' parameter on Captive Portal pages.
- Added output encoding to diag_dns.php for results returned from DNS. #6737
- Worked around a Chrome bug with regular expression parsing of escaped characters within character sets. Fixes "Please match the requested format" on recent Chrome versions. #6762
- Fixed DHCPv6 server time format option #6640
- Fixed /usr/bin/install missing from new installations. #6643
- Increased filtering tail limit for logging so searching will locate sufficient entries. #6652
- Cleaned up Installed Packages widget and HTML. #6601
- Fixed widget settings corruption when creating new settings. #6669
- Fixed various typos and wording errors.
- Removed defunct links to the devwiki site. Everything is on https://www.netgate.com/docs/pfsense/ now.
- Added a field to CA/Cert pages for OU, which is required by some external CAs and users. #6672
- Fixed a redundant HTTP "User-Agent" string in DynDNS updates.
- Fixed the font for sortable tables.
- Added a check to verify if an interface is active in a gateway group before updating dynamic DNS.
- Fixed wording of the "Reject leases from" option for a DHCP interface (it can only take addresses, not subnets.) #6646
- Fixed error reporting for SMTP settings test.
- Fixed saving of country, provider, and plan values for PPP interfaces
- Fixed checking of invalid "Go To Line" numbers on diag_edit.php. #6704
- Fixed off-by-one error with "Rows to Display" on diag_routes.php. #6705
- Fixed description of the filter box on diag_routes.php to reflect that all fields are searchable. #6706
- Fixed description of the box for the file to edit on diag_edit.php. #6703
- Fixed description of the main panel on diag_resetstate.php. #6709
- Fixed warning dialog when a box is unchecked on diag_resetstate.php. #6710
- Fixed log shortcut for DHCP6 areas. #6700
- Fixed the network delete button showing when only one row was present on services_unbound_acls.php #6716
- Fixed disappearing help text on repeatable rows when the last row is deleted. #6716
- Fixed dynamic DNS domain for static map DHCP entries
- Added control to set dashboard widget refresh period
- Added "-C /dev/null" to the dnsmasq command line parameters to avoid it picking up an incorrect default configuration which would override our options. #6730
- Added "-l" to traceroute6 to show both IP Addresses and Hostnames when resolving hops on diag_traceroute.php. #6715
- Added note about max ttl/hop limit in source comment on diag_traceroute.php.

- Clarified language on diag_tables.php. #6713
- Cleaned up the text on diag_sockets.php. #6708
- Fixed display of VLAN interface names during console assignment. #6724
- Fixed domain-name-servers option showing twice in pools when set manually.
- Fixed handling of DHCP options in pools other than the main range. #6720
- Fixed missing hostnames in some cases with dhcpdv6. #6589
- Improved pidfile handling for dhcpleases.
- Added checks to prevent accessing an undefined offset in IPv6.inc.
- Fixed the display of the alias popup and edit options on source and destination for both the address and port on outbound NAT.
- Fixed handling of backup config count. #6771
- Removed some dangling PPTP references that are no longer relevant.
- Fixed up/caught up remote syslog areas. Added "routing", "ntpd", "ppp", "resolver", fixed "vpn" to include all VPN areas (IPsec, OpenVPN, L2TP, PPPoE Server). #6780
- Fixed missing checkboxes in some cases when adding rows on services_ntpd.php. #6788
- Revised service running/stopped icons.
- Added a check to CRL management to remove certificates from the drop-down list that are already contained in the CRL being edited.
- Fixed rule separators moving when multiple firewall rules are deleted at the same time. #6801

3.3.20 2.3.2 New Features and Changes

SSH Daemon

NOTE: The ssh host keys were made more secure, and if a client remembers an older, weaker key, the ssh client may refuse to connect. Remove the older key and then make the ssh client learn the new key.

- Changed sshd to use stronger Key Exchange algorithms and disabled some older, weaker algorithms. Clients may need to be updated to handle the new Key Exchange methods.
  - Currently allowed Key Exchange Algorithms: curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
- Removed the ECDSA host key from the sshd configuration
- Added ED25519 host key to the sshd configuration
- Changed the list of available ciphers.
  - Current allowed ciphers: chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
- Changed the list of available Message Authentication Code methods,
  - Current MAC list: hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-ripemd160-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-ripemd160,umac-128@openssh.com

Backup/Restore

- Don't allow applying changes on interface mismatch post-config restore until the reassignment is saved. #6613

Dashboard

- Dashboard now has per-user configuration options, documented in *User Manager*. #6388

DHCP Server

- Disabled dhcp-cache-threshold to avoid bug in ISC dhcpd 4.3.x omitting client-hostname from leases file, which makes dynamic hostname registration fail in some edge cases. #6589
- Note that DDNS key must be HMAC-MD5. #6622

DHCP Relay

- Imported fix for dhcrelay relaying requests on the interface where the target DHCP server resides. #6355

Dynamic DNS

- Allow * for hostname with Namecheap. #6260

Interfaces

- Fix "can't assign requested address" during boot with track6 interfaces. #6317
- Remove deprecated link options from GRE and gif. #6586, #6587
- Obey "Reject leases from" when DHCP "Advanced options" is checked. #6595
- Protect enclosed delimiters in DHCP client advanced configuration, so commas can be used there. #6548
- Fix default route on PPPoE interfaces missing in some edge cases. #6495

IPsec

- strongSwan upgraded to 5.5.0.
- Include aggressive in ipsec.conf where IKE mode auto is selected. #6513

Gateway Monitoring

- Fixed "socket name too large" making gateway monitoring fail on long interface names and IPv6 addresses. #6505

Limiters

- Set pipe_slot_limit automatically to maximum configured qlimit value. #6553

Monitoring

- Fixed no data periods being reported as 0, skewing averages. #6334
- Fix tooltip showing as "none" for some values. #6044
- Fix saving of some default configuration options. #6402
- Fix X axis ticks not responding to resolution for custom time periods. #6464

OpenVPN

- Re-sync client specific configurations after save of OpenVPN server instances to ensure their settings reflect the current server configuration. #6139

Operating System

- Fixed pf fragment states not being purged, triggering "PF frag entries limit reached". #6499
- Set core file location so they can't end up in /var/run and exhaust its available space. #6510
- Fixed "runtime went backwards" log spam in Hyper-V. #6446
- Fixed traceroute6 hang with non-responding hop in path. #3069
- Added symlink /var/run/dmesg.boot for vm-bhyve. #6573
- Set net.isr.dispatch=direct on 32 bit systems with IPsec enabled to prevent crash when accessing services on the host itself via VPN. #4754

Router Advertisements

- Added configuration fields for minimum and maximum router advertisement intervals and router lifetime. #6533

Routing

- Fixed static routes with IPv6 link local target router to include interface scope. #6506

Rules / NAT

- Fixed "PPPoE Clients" placeholder in rules and NAT, and ruleset error when using floating rules specifying PPPoE server. #6597
- Fixed failure to load ruleset with URL Table aliases where empty file specified. #6181
- Fixed TFTP proxy with xinetd. #6315

Upgrade

- Fixed nanobsd upgrade failures where DNS Forwarder/Resolver not bound to localhost. #6557

Virtual IPs

- Fixed performance problems with large numbers of virtual IPs. #6515
- Fixed PHP memory exhaustion on CARP status page with large state tables. #6364

Web Interface

- Added sorting to DHCP static mappings table. #6504
- Fixed file upload of NTP leap seconds. #6590
- Added IPv6 support to diag_dns.php. #6561
- Added IPv6 support to filter logs reverse lookup. #6585
- Package system - retain field data on input error. #6577
- Fixed multiple IPv6 input validation issues allowing invalid IPv6 IPs. #6551, #6552
- Fixed some DHCPv6 leases missing from GUI leases display. #6543
- Fixed state killing for 'in' direction and states with translated destination. #6530, #6531
- Restore input validation of captive portal zone names to prevent invalid XML. #6514
- Replaced calendar date picker in the user manager with one that works in browsers other than Chrome and Opera. #6516
- Restored proxy port field to OpenVPN client. #6372
- Clarify description of ports aliases. #6523
- Fixed translation output where gettext passed an empty string. #6394
- Fixed speed selection for 9600 in NTP GPS configuration. #6416
- Only allow IPv6 IPs on NPT screen. #6498
- Add alias import support for networks and ports. #6582
- Fixed sortable table header wrap oddities. #6074
- Clean up Network Booting section of DHCP Server screen. #6050
- Fix "UNKNOWN" links in package manager. #6617
- Fix missing bandwidth field for traffic shaper CBQ queues. #6437

UPnP

- UPnP presentation URL and model number now configurable. #6002

User Manager

- Prohibit admins from deleting their own accounts in the user manager. #6450

Other

- Added PHP shell sessions to enable and disable persistent CARP maintenance mode. "playback enablecarpmaint" and "playback disablecarpmaint". #6560
- Exposed serial console configuration for nanobsd VGA. #6291

3.3.21 2.3.1 New Features and Changes

Security/Errata

- FreeBSD Security Advisories
  - FreeBSD-SA-16:17 OpenSSL
  - FreeBSD-SA-16:18 atkbd
  - FreeBSD-SA-16:19 sendmsg
- OpenVPN upgraded from 2.3.10 to 2.3.11. Fixes two potential security issues.
  - OpenVPN 2.3.11 Change Log
- pfSense® Software Advisories
  - pfSense-SA-16_03.webgui
  - pfSense-SA-16_04.filterlog
  - 2.3.1 update 1 patches pfSense-SA-16_05.webgui.

Config Upgrade

- Fixed config upgrade for CARP VIPs on gateway groups, GRE and gif for uniqid format. #6222
- Fixed config upgrade for IP aliases with CARP IP parent. #6164
- Correct OpenVPN topology config upgrade to retain 2.2.x and prior net30 topology. #6140
- Correct and adjust apinger parameters to dpinger parameters automatically on upgrade. #6142

Gateways

- Fix static route for IPv6 monitor IP with link-local gateway. #6353
- Fix default gateway switching with IPv6 and link-local gateways. #6258

OS / Backend

- NanoBSD is now permanent read-write, to avoid issues with slow rw->ro mount times and systems getting stuck read-only mounted. #6184
- Systems using a RAM disk for /var/ have their alias tables backed up and restored during bootup. #6189
- Set console settings (serial configuration, password protection, etc.) post-upgrade. #6120
- Ensure package repo is updated with latest metadata when checking for latest version. #6115
- Display consistent firmware version on dashboard and in update checker. #6320
- Correct description of update branch options. #6136
- Prevent update checking failures from killing webGUI. #6177
- Make pkg use configured proxy server settings where they exist. #6149

Web GUI

- Fix row delete button on unsaved aliases, NTP, UPnP and other screens. #6101
- Captive portal MAC passthrough credits waiting period box restored. #6290
- Outbound NAT edit screen destination field alias auto-completion restored. #6287
- Captive portal allowed IPs direction selection on edit fixed. #6267
- Restored input validation on port forwards to prohibit IPv6. #6265
- Restored input validation on firewall rules to prohibit IPv6 IPs in IPv4 rules and vice versa. #6211
- Fixed PHP error on edit of PPP interfaces. #6264
- Fixed radio button placement on gateways dashboard widget settings. #6259
- Fixed display post-refresh of system information dashboard widget. #6251
- Restored in/out bytes counters on Status>Interfaces. #6244
- Correctly show and hide OpenVPN topology field as applicable. #6236 #6214
- Correct voucher character set input validation. #6231
- Disable background update checking on dashboard update check is disabled. #6212
- Restore input validation of IP address family and rule type, verifying IPv6 IPs with IPv6 rules, and IPv4 for IPv4 rules. #6218
- Add validation of address family and protocol combinations on packet capture page. #6219
- Add validation of IP aliases with CARP parent interfaces to ensure matching address family. #6218
- Restore GET parameters on status_graph.php. #6192
- Fixed PHP error on input validation failure with floating rules in some cases. #6175
- Use CDATA for firewall rule separator descriptions so non-English characters work. #6174

- Fix port forward edit destination field filling when virtual IPs configured. #6173
- Fix load balancer monitor edit. #6171
- Restore "none" in load balancer fall-back pool. #6170
- Restore use of aliases in load balancer. #6169
- Fix duplicate for load balancer pools and virtual servers. #6168
- Restore description field on lagg edit page. #6163
- Fix saving of bogons update frequency. #6162
- Restore description field on captive portal IP passthrough. #6161
- Fix saving of sticky connections timeout field. #6146
- Show all restore areas in backup/restore screen. #6144
- Fix moving of rule separator before saving. #6128
- Use consistent up and down arrow formats on dashboard widgets. #6123
- Fix typo on OpenVPN server description. #6102
- Fix missing string on notification "mark as read" button. #6104
- Fix firewall rule separator positioning with easy rule addition. #6105
- Prevent closing of info box on monitoring page. #6106
- Add custom date range option to monitoring page.
- Use infoblock on IPsec PSK screen. #6107
- Fixed loss of "Do not NAT" enable on edit on outbound NAT. #6112
- Correct label of 1:1 NAT edit screen. #6114
- Add AJAX updates to NTP status page. #6117
- Fix button spacing on Edit File and Command pages. #5995
- Fix specification of port in DNS Resolver domain overrides. #6091
- Fix moving of multiple items to bottom of list on firewall, NAT and IPsec screens. #6092
- Fix setup wizard with only WAN assigned and using static IP. #6093
- Remove logo from wizard since it's now redundant. #6095
- Fix gateway widget cut-off with 3 column dashboard. #6096
- Fixed force update on RFC 2136 DDNS. https://redmine.pfsense.org/issues/6359
- Fix reboot prompt when changing RAM disk setting and encountering an input error. #6349
- Fix highlighted tab when editing IPsec mobile P1. #6341
- Fix selection of configured speed and duplex on interface page. #6331
- Fix division by zero in status_queues.php. #6329
- Fix alignment issues in forms. #6327
- Fix entry of CIDR range in host aliases for conversion to IPs. #6322
- Allow use of # and ! again in DNS Forwarder domain overrides. #6310
- Restored hostname infobox in menu bar. #6306

- Fixed editing and deleting of additional DHCP pools. #6303
- Fixed requests to diag_system_activity.php piling up on slow systems. #6166

Interfaces

- Unset LAN DHCPv6/RA configuration if LAN interface is removed. #6152

IPsec

- Fix starting of strongswan twice. #6160

DNS Resolver

- Switched domain overrides from stub-zone to forward-zone so domain overrides don't require the target server provide recursion. #6065
- Allow adding 0.0.0.0/0 to access lists. #6073
- Added 100,000 and 200,000 options for Unbound cache limit. #6230
- Fix Unbound startup where both DNS Forwarder and Resolver are enabled. #6354

DHCP Server

- Hostnames now allowed for NTP servers. #6239

IPsec

- Fixed LAN interfaces stopping functioning when IPsec is in use. #6296
- Mobile PSK matching issue with multiple PSKs fixed. #6286
- leftsendcert=always specified for all RSA types. #6082
- rc.newipsecdns fixed to check correct enabled status. #6351

Notifications

- Fixed growl notifications to unresolvable hostname generating crash report. #6187
- Fixed growl notification test with no password. #6221

Captive Portal

- Fixed error handling captive portal username with single quote. #6203
- Fixed issues with mixed-case zone names. #6278

OpenVPN

- Prevent leading space in tunnel network configuration causing invalid configuration. #6198

User Manager

- Fix RADIUS login with attribute class (25) when the server returns multiple attribute entries with different data. #6086
- Honor deny config write for RADIUS users. #6088

Package System

- Uninstall all packages pre-upgrade from <= 2.2.x to 2.3 to avoid problems from old packages. Reinstall them post-upgrade. #6137
- Fix reinstall of renamed packages post-upgrade to 2.3. #6118
- Fix package reinstallation getting stuck in loop when there is no Internet connectivity post-upgrade. #6180

Other

- Removed lua support from nginx to not deprecate old CPUs lacking CMOV support. #6185
- Added validation to console menu interface assignment to prevent creating duplicate VLANs. #6183
- Blacklisted S.M.A.R.T. options with Hyper-V to prevent crash. #6147
- Silence SSH host key log spam. #6143
- Fix order of gateway and gateway group name in gateway down log message. #6134
- Allow use of @ in hostname field for Namecheap DDNS. #6122
- Fix console error where $nat_if_list isn't an array. #6307
- Include patch number in version display. #6309
- Fix pw groupdel error in log during boot. #6352
- Fixed stale xmlrpc.lock preventing config sync from functioning. #6328
- Fixed failed chown on startup with /var as a RAM disk. #6131
- Crash reporter now ignores warnings in release versions. #6178
- Fixed crash reporter to show full PHP warnings in development versions. #6097

Update 1

2.3.1 update 1 (2.3.1_1) was released on May 25, 2016 with the following fixes/changes since 2.3.1-RELEASE.

- Security issue pfSense-SA-16_05.webgui patched.
- Lowered default LDAP timeout from 25 seconds to 5 seconds. #6367
- Fixed handling of IPsec negotiation mode with IKE version set to auto. #6360
- Increase PHP's memory limit to 512 MB on 64 bit versions to better accommodate systems with a large number of active states. #6364

- Set request_terminate_timeout the same as max_execution_time to prevent many possible circumstances of "504 gateway error" from occurring. #6396
- Fix use of URL IP type aliases in firewall rules. #6403
- Fix show/hide fields Javascript in Chrome on Mac OS X. #6401
- Fixed save of "IPv6 over IPv4 Tunneling" address on System>Advanced, Networking. #6381

Update 2 through 4

These were internal-only versions that weren't publicly-released.

Update 5

2.3.1 update 5 (2.3.1_5) was released on June 16, 2016 with the following fixes/changes since 2.3.1_1.

- Fixed command injection vulnerability in auth.inc via User Manager. #6475
- Fixed command injection vulnerability in pkg_mgr_install.php id parameter. #6474
- Upgraded PHP to 5.6.22
- Fixed Captive Portal redirect hangs caused by longer keepalive_timeout in nginx. #6421
- Fixed DDNS PTR zone in dhcpd.conf with third octet of 0. #6413
- Fixed save and reset buttons on load balancer status page. #6254
- Fixed schedule editing on firewall rules page. #6428
- Allow "-" character in TFTP server field on DHCP Server page. #6433
- Allow "-" and "_" characters in system tunables. #6438
- Fixed changing of link type on PPPs edit screen. #6439
- Fixed setting of "RADIUS issued IPs" on L2TP page. #6440
- Restored apply changes button for interface mismatch post-config restore. #6460
- Fixed display of Outbound NAT port aliases. #6463
- Fixed schedule edit allowing invalid time range. #6468

3.3.22 2.3 New Features and Changes

Security/Errata

- FreeBSD Security Advisories:
  - FreeBSD-SA-16:01.sctp
  - FreeBSD-SA-16:02.ntp
  - FreeBSD-SA-16:05.tcp
  - FreeBSD-SA-16:07.openssh
  - FreeBSD-SA-16:09.ntp
  - FreeBSD-SA-16:11.openssl

  - FreeBSD-SA-16:12.openssl
  - FreeBSD-SA-16:15.sysarch
- pfSense® Security Advisories:
  - pfSense-SA-16_01.webgui
  - pfSense-SA-16_02.webgui

Several obsolete items were removed from this release. The items are noted again in the sections below, but worth emphasizing:

- The PPTP **VPN Server** has been completely removed. The protocol has been broken for over three years.
  - The PPTP WAN client remains for use with ISPs still using PPTP.
- Layer 7 classification support has been removed from the traffic shaper.
  - It was rarely used, had been broken for all of 2.2.x, had absurdly high CPU usage, and snort filters better/faster
- WEP support has been removed from Wireless interfaces. #5123
  - No reason to still be using this in this day and age. If it is still needed, use external AP.
- Single DES support has been removed from IPsec (3DES remains).
  - It should not be used, it is not secure.
- 1GB NanoBSD images have been removed, as they were not large enough to proper accommodate the system and upgrade data. The supported sizes for NanoBSD images are now 2GB and 4GB.
- The default system password hash has been changed to bcrypt. Current passwords will continue to work. Existing users need to reset their password to convert to the new hash. More info below under "Authentication". #4120
- The LiveCD platform has been removed. The ISO is a bootable installer, as always, but it cannot run a live system.
  - The installer ISO image is now named "pfSense–RELEASE-.iso", with the .iso extension signifying the type of image it is (optical media installer).
  - For the very few people who were still using LiveCD, if the hardware can boot from USB, install to a USB thumb drive and run from it instead. If the options to keep /var and /tmp in RAM are active, and no packages are installed, the net result should be similar but ultimately more functional.

Dashboard/Widgets/GUI

- Converted GUI to the Bootstrap framework, completely new look
- Changed the GUI and Captive Portal web server to nginx; removed lighttpd. #5719
- Cleaned up a lot of GUI code, option text, etc
- TLS v1.0 disabled for the GUI. #5984
- Removed old style themes, introduced new CSS-based themes
- Refactored JavaScript and CSS, moved included items to more convenient locations
- Added more AJAX updating in widgets and other places
- Changed to more intuitive and modern icons and action buttons rather than the old confusing icon set (now using font-awesome icons)

- Changed log display to be more consistent (single page for most logs, common filtering options)
- Removed obsolete fifolog support. It was never used or fully implemented, and had no GUI option.
- Improved notices in the GUI
- Made breadcrumbs and page title handling more consistent
- Added an option to have the top menu follows the user when scrolling
- Renamed several GUI file names to match menu structure. #5628
- Fixed AES-NI hardware display in the system information widget. #4911
- Added widescreen support to the Dashboard. #5195
- Improved password field handling security. Stored passwords are not presented back to the user in HTML. A masked value is returned instead. All password fields have also been changed to require confirmation.
- Many pages have been reworked for improved internationalization
- Changed info box functions, removed print_info_box_np, now print_info_box and print_apply_box are used to print appropriate boxes without problematic automatic detection
- Moved RRD graphs to **Status > Monitoring** #5498
- Changed RRD GUI interface to D3 rather than using the RRD graph command, so that a newer rrdtool base could be used with minimal added dependencies. #5498
- Monitor IP added to gateways widget. #4782
- Increased max_input_vars from 1000 to 5000 to accommodate larger aliases. #4780
- Fixed NTP RRD graphs to accommodate negative values. #4423

OS/Backend

- Moved to a FreeBSD 10.3-RELEASE base
- Added tryforward() support to get (nearly all of) the performance of fastforward with IPsec enabled
- Overhauled the build system
  - Eliminated the -tools repository
  - Removed Patches, changes are now applied a vendor branch of FreeBSD
  - Rewrote/changed the build scripts significantly
  - Moved the new build scripts to the main pfSense repository
- PHP Upgraded to 5.6
- Replaced pecl-APC with opcache. #4744
- Added support for -c parameters to /etc/rc.initial. #4422
- Added optional package for kernel debug symbols. #5330
- Rewrote system_set_harddisk_standby() for the current CAM-based ATA stack. #4569
- Fixed a Panic/Crash with “sbflush_internal: cc 4294967166 || mb 0 || mbcnt 0”. #4689
- Fixed a kernel panic with AES-NI. #4702
- Updated AES-GCM/AES-NI bits from FreeBSD -HEAD. #4841
- Removed zoneinfo.tgz file for Time Zones, move to the same format as FreeBSD. #4726

- Fixed tcpdump with zerocopy enabled (net.bpf.zerocopy_enable=1). #5257
- Added ability to disable PV disks and NICs on Xen. #5452
- Removed the built-in but unused MySQL PHP modules and added them to the pkg server instead. They may be added as package dependencies or manually installed as needed.
- Followed FreeBSD (r294560) in ceasing generation of rsa1 and dsa ssh server host keys by default
- Removed support for nanobsd images < 2GB #5836
- Overhauled IP address handling code in various parts of the system
- scponly package is included by default. #5190
- Shortened F1 boot prompt delay on nanobsd. #3426

Packages

- The list of available packages in pfSense 2.3 has been significantly trimmed. We have removed packages that have been deprecated upstream, no longer have an active maintainer, or were never stable.
- Removed use of PBI-based packages, moved to pkg(ng)
- Fixed installation and handling of packages to use pkg, now works identically in the GUI and shell/console
- Changed packages to use the FreeBSD ports format/layout to work with pkg
- XMLRPC calls for package information and installation have been removed, replaced with native pkg functions. #4575
- Added support for packages to be (re)built automatically by Poudriere
- Added search capability to Available Packages list to filter packages by keywords. #5324
- Fixed the version comparison code in the Package manager. #4924
- Added support for tags in listtopic fields for use by packages
- Factory reset now completely uninstalls packages. #5829
- Improved handling of package install post-upgrade. #3597

System Updates

- Major changes to update management
- Removed “full update” or “full slice” upgrade for systems on 2.3 to later versions
  - These files will remain available for use by older versions updating to 2.3.
- The “Full Backup” feature has been deprecated.
- Changed system updates to be handled via pkg
- Changed Base, kernel, and standard pre-installed binares to packages
- Removed “Firmware” nomenclature, now only referred to as “Update”
- Fixed updating of base to work the same from the console or the GUI
- Added preliminary support for restarting system services without rebooting in cases when the base is updated but the kernel is the same.

Gateways/Routing

- Replaced apinger with dpinger(!). #5624
  - This fixes many gateway monitoring related issues, including incorrect latency and loss in various edge cases.
  - Eliminates status file race conditions that caused update failures on services bound to gateway groups in some edge cases. #5180 and #3818 among others.
  - Fixed gateway monitoring startup at boot time with assigned OpenVPN interfaces. #4587
  - Check gateway monitor settings after upgrade, dpinger has different options than apinger.
- Added code to allow gateways outside of an interface subnet. #972
- Corrected "State Killing on Gateway Failure" description. #4709
- Fixed disabling of a static route set to use a disabled gateway. #4813
- Added standard deviation to gateway status and widget
- Fixed dynamic gateway logic to prevent GIF/GRE from making dummy/unusable gateways that show up for monitoring/routing/etc #5766
- Changed static routes handling for DNS servers so they are removed when a gateway is disabled #4921
- Increased gateway weight limit from 5 to 30. #5843
- Fixed issues with PPP type WANs and the Default Gateway Switching option. #1837
- Fixed dynamic gateway handling for OpenVPN tap clients. #5981
- Fixed display of full interface name in Diagnostics>Routes. #5484

Rules/NAT/pf

- Added drag-and-drop rule reordering for firewall and NAT rules.
- Fixed a situation where pf drops IPv6 packets with fragment header followed by a last fragment only. #2762
- Fixed "LAN network" in v6 rules not working when a link-local address is assigned to LAN. #3656
- Added reordering for 1:1 NAT rules. #3888
- Improved handling of firewall rule tracker IDs for port forward associated rules
- Added support for a separator bar in firewall and NAT rules for use as a visual reference. #5373
- Standardized the NPt options in the GUI so their options and appearance are more similar to 1:1 NAT
- Added a "no binat" checkbox to 1:1 NAT screen for exclusions. #3887
- Limited pfsync syncpeer to IPv4 since it does not support IPv6 #4648
- Changed the default CARP pass rules to use "no state" to avoid issues with broken L2 gear that duplicates packets #5800
- Added sorting to Alias lists #4195
- Added a hit counter to the firewall rule display with states and bandwidth consumed by packets matching rules.
- Fixed issues with the DNS Forwarder and DNS Resolver being enabled concurrently (on different ports) in an HA environment #5882
- Added a visual indication in the rule list for floating rules with the "quick" property set #5860

- Improved state display on **Diagnostics > States**, now shows packets and bytes for each state
- Fixed aliases containing both FQDNs and IPv6 subnets. #5872
- Fixed removal of downloaded URL table alias contents when alias is deleted. #5856
- Significantly improved validation of downloaded data for URL Table aliases. #5848
- Fixed possibilities for creating an invalid ruleset with missing URL Table Ports aliases. #5845
- Fixed filterdns issues with significant system clock time jumps. #4166
- Added firewall rules hit counter. #3504

Interfaces/VIPs

- Fixed pfSense_getall_interface_addresses truncating IPv6 link local IP addresses. #4062
- Add GUI setting for VLANs PCP. #4133
- Fixed GRE interfaces failing to have a RUNNING state after reboot. #4191
- Fixed setting non-default MTUs in some edge cases. #4397
- Added input validation on bridges to prevent adding the same interface to multiple bridges. #4595
- Fixed CARP not working under bhyve. #4623
- Improved input validation for 6RD, GRE and gif interfaces, helping prevent invalid configurations.
- Changed input validation to allow /31 to be used for CARP VIPs since that is now supported and works in FreeBSD. #5533
- Added debug logging option for DHCP6 client. #4534
- Fixed cases where DHCP6 client (dhcp6c) was being launched multiple times in some circumstances. #5621
- Upgraded dhcp6c. #5734
- Upgraded DHCP client to ISC dhcpd 4.3.3P1.
- Fixed applying of non-default MTU on gif interfaces post-boot with dynamic IP WANs. #5842
- Added support for PPPoE with MTU/MRU > 1492, RFC 4638. #4542
- Fixed issues with link cycling on some Intel 10G ix NICs #5913
- Corrected ALTQ test to show that ix/ixgbe NICs are capable of traffic shaping. #5923
- Improved handling of default interface assignment for some hardware. #4535
- Corrected input validation for invalid IPv6 IPs with leading or trailing colon. #6024
- Fixed orphaning of VLANs on lagg interfaces after editing the lagg. #6014
- Fixed loss of some dhcpleases and dhcpleases6 logs. #5968
- Fixed adding of routes immediately post-reboot for delegated IPv6 prefixes to sub-routers. #5957
- Fixes to DHCPv6 leases status page and prefixes.php. #5944 #4206
- Fixed loss of IPv6 IP on track6 interfaces when saving and applying changes on that interface. #5945
- Fixed incorrect interface mismatch prompt post-config restore when using VLANs on lagg. #5892
- Added support for multiple span interfaces on bridges. #5871
- Prevent naming conflicts between interfaces and interface groups. #5795

- Prevent naming conflicts between interfaces and aliases. #5778
- Fixed use of IP aliases with GRE tunnels. #4450
- Fixed application of bridge advanced options after interface added to bridge. #4312
- Set MTU back to default after clearing the field. #3926
- Fixed IPv6 IP aliases on CARP IPs. #3716
- Fixed IP alias on CARP IPs where IP alias above CARP parent in list. #3257
- Fixed modifying unassigned VLAN interfaces changing assigned VLAN. #3209

Authentication

- Fixed the WebGUI becoming slow or unusable when an LDAP server used for GUI auth is unreachable. #3383
- Fixed a problem with using 'local' as the name of an authentication server 'Descriptive Name'. #4469
- Fixed default Auth Server selection on system_usermanager_settings.php. #5440
- Added support for bcrypt as a passwd hash and enabled it as the system default #4120
- Replaced the default passwd hash for root/admin using bcrypt (blowfish).
  - Existing user passwords will continue to work in their existing format until the user's password is changed.
  - User passwords cannot be automatically converted as they are not stored plain text. To convert the password hash of an existing user to bcrypt, edit the user and change their password.
- Added the ability to filter privileges when adding them to a user or group, to make finding them easier.
- Fixed updating of group file for renamed groups. #6013
- Fixed handling of groups with spaces in their names. Local group names can no longer contain spaces. New group scope option "Remote" added for LDAP and RADIUS use where spaces in group names are valid. #6012
- Added support for RFC2307 style LDAP groups. #4923

Services

- Fixed handling of the SNMP Bind Interface. #3883
- Fixed ntpd crashes on 32 bit with dynamic WAN reconnections and OpenVPN client configured. #4155
- Fixed a kernel panic with APU and SNMP with mibII. #4403
- Updated igmpproxy to the latest version. #4672
  - The old version had some custom patches, so be wary of behavior changes
- Added encoding for DHCP/DHCPv6 server additional BOOTP text options to preserve data when stored in XML #5623
- Fixed duplication action for Load Balancer Monitor entries #4441
- Upgraded DHCP Server and Relay to ISC dhcpd 4.3.3P1
- Added statistics gathering for DHCP Server leases. #5387
- Fixed DDNS key issues with DHCP and DHCPv6 Server enabled on multiple interfaces. #5603
- Added custom ACLs for NTP (restrictions by network) #4463
- Prevent starting of radvd in circumstances where it shouldn't. #5812

- Added description column to DHCP leases status screen. #5729
- inetd replaced with xinetd (used for proxy mode NAT reflection and TFTP proxy). #5707
- DHCP lease counters added to Status>DHCP Leases. #5186
- Allow configuration of RAs when DHCPv6 Relay is enabled. #6063
- Fixed DHCPv6 Server's DDNS. #4675
- DHCP Server menu item now defaults to the first interface with an enabled DHCP Server instance. #4647
- Allow configuring DHCPv6 and RAs on track6 interfaces. #3029
- Fixed RADIUS NAS IP in PPPoE server. #185
- Deprecated ntpdate_sync_once.sh, replacing with ntpd -g. #6053

DNS

- Fixed Unbound IPv6 link local handling. #4021
- Added validation for advanced configuration directives in Unbound. #4411
- Upgraded dnsmasq to 2.76.0test8 to fix crashes in 2.75. #5341
- Fixed Unbound binding to IP alias virtual IPs. #5464
- Changed Namecheap dynamic DNS to use separate hostname and domain name fields #4366
- Added Multi-WAN support to RFC 2136 Dynamic DNS.
- Added RFC 2136 support to the Dynamic DNS widget
- Added input validation to prevent the same DNS server from being added multiple times on **System > General** #5915
- Fixed CloudFlare dynamic DNS to not configure 'proxiable' and 'proxied' parameters. #6005
- Fixed dnsmasq host overrides when both DNS Forwarder and Resolver are enabled. #5883
- Added RFC 2136 dynamic DNS to dashboard widget. #5862
- Added multi-WAN support to RFC 2136 dynamic DNS client. #5862
- Don't specify 127.0.0.0/8 IPs as forward-addr in Unbound configuration. #5750
- Added input validation to require configured DNS servers before enabling Resolver's forwarding mode. #4747
- Added Google Domains DDNS support. #4322
- Added DNS Made Easy DDNS support. #1258
- Allow @ in Dynamic DNS hostnames. #3900
- Improve IPv6 link local handling in DNS Resolver and Forwarder so it works across configuration restores and with HA config sync. #3802

IPsec

- Upgraded to strongSwan 5.4.0.
- Fixed multiple possibilities for IPsec status hangs. #5520
- Revised handling of IPsec reloading when strongswan.conf is changed. #4353
- Fixed problems with the search domain in IPsec mobile clients. #4418
- Added support for elliptic curve for IPsec on webconfigurator. #4683
- Added input validation for authentication backend when using EAP-RADIUS with IKEv2 Mobile IPsec. #5219
- Fixed unit display on IPsec status pages for time and data to be more human-friendly. #5364
- Removed support for single DES from IPsec #5543 (3DES Remains)
- Removed global IPsec disable flag as it is no longer necessary. On upgrade, if the IPsec enable box was unchecked, all Phase 1 entries are disabled individually instead.
- Changed IPsec 'up' commands to start in the backgound so they are non-blocking #5882
- Disabled the strongSwan unity plugin by default, and improved the method used to disable the plugin #4178
- Removed unnecessary and troublesome 'pass out' rules for mobile IPsec #5819
- Fixed "no valid leases object found" log spam with IPsec dashboard widget. #5855
- Fixed automatically added WAN rules (UDP 500, 4500, ESP) when using IPsec with IP aliases. #5500
- Fixed IKEv2 to Cisco ASA resulting in traffic selector mismatch when initiated by traffic. #4719
- Added "split connections" option to phase 1 for IKEv2 for interoperability with third party devices that do not support multiple traffic selectors on one child SA (Cisco ASA, others). #4704
- Added dynamic AJAX update to status_ipsec.php. #6049

OpenVPN

- Changed the default behavior of the OpenVPN server to use topology subnet, not net30. #5526
- Changed Client-Specific Overrides so they can be set to apply to specific servers rather than being globally set. #5526
- Fixed OpenVPN Server validation of self-signed certificates with a depth of 2. #4329
- Fixed overwriting of custom `/etc/dh-parameters.*` on upgrade. #4816
- Fixed invalid rules generated with some AVPair-defined ACLs. #5451
- Improved display of server certificates on OpenVPN servers to help avoid users incorrectly picking user certificates for servers. #5602
- Fixed OpenVPN client specification of auth-user-pass in shared key modes where it's not valid. #5941
- Fixed problems with OpenVPN and some use of special characters in the username or password. #4605

MPD/PPP VPN/Services

- Removed PPTP Server. #4226
- Add MS-CHAPv2 option to L2TP Configuration. #4732
- Fixed editing of multiple PPPoE connections with dial on demand enabled changing the port assignment. #4378
- Added a user login count option to the PPPoE server

UPnP/NAT-PMP

- Enabled port-in-use checking in miniupnpd. #4320
- Enabled IPv6 for miniupnpd. #4321
- Set secure_mode=yes in miniupnpd configuration #5627

Wireless

- Removed WEP. #5123
- Improved default settings for Wireless interfaces

Captive Portal

- Fixed Captive Portal to support more than 120 VLAN interfaces. #4150
- Added an option in Captive Portal for FreeRADIUS-friendly stop/start RADIUS accounting updates that solves problems with user session time limits. #2164
- Fixed selection of RADIUS NAS IP with VIPs when editing Captive Portal zone. #5656

Traffic Shaping

- Fixed CODELQ scheduler defaults. #4692
- Removed Layer 7 classification support from the traffic shaper #5508
- Relaxed the shaper wizard interface validation when there are no interfaces with gateways selected #4524
- Fixed traffic shaper failure with "bandwidth for q… higher than interface" in some edge cases. #5721

Misc

- Allow wildcards in Certificate Subject Alternative Names. #3733
- Removed the "Certificate Authority" option on the **Certificates** tab of the Cert Manager when creating a **Certificate**. To make a Certificate Authority, use the **CAs** tab instead. #5924
- Adapted gitsync to new repo structure. #4999
- Changed the packet capture output in the GUI so that when the protocol is set for CARP, tcpdump interprets it as CARP for more accurate output
- Added pfsync protocol option to packet capture page. #5866
- Added "GoTo line #" control to Diagnostics > Edit File

- Corrected help in pfSsh.php to properly reflect how recording works
- Fixed validation of playback file passed to pfSsh.php #5657
- Fixed disabling of filter.log logging where local logging is disabled. #6018
- Updated included software on licenses.php page. #5903
- Internationalization improvements. #5777
- Fixed use of IP aliases on Test Port page. #5185
- Fixed key map, screen map and font selection in installer. #4387
- Prevent deletion of certificates in use by packages. #4142

Update Patches

This section lists the changes contained in patch updates post-release.

2.3_1

The 2.3_1 update upgrades NTP to fix FreeBSD security advisory SA-16:16.ntp. The only change is upgrading ntpd from 4.2.8p6 to 4.2.8p7.

3.3.23 2.2.6 New Features and Changes

Security/Errata Notices

- Updated to FreeBSD 10.1-RELEASE-p25
  - FreeBSD-SA-15:26.openssl Multiple vulnerabilities in OpenSSL
- Updated to strongSwan 5.3.5
  - Includes fix for CVE-2015-8023 authentication bypass vulnerability in the eap-mschapv2 plugin.
- pfSense-SA-15_09.webgui: Local File Inclusion Vulnerability in the pfSense® WebGUI
- pfSense-SA-15_10.captiveportal: SQL Injection Vulnerability in the pfSense captive portal logout
- pfSense-SA-15_11.webgui: Multiple XSS and CSRF Vulnerabilities in the pfSense WebGUI

Logging

- Fixed log duplication for some log entries. #5606

IPsec

- strongSwan 5.3.5 update fixes several bugs.

Config sync

- Fixed config synchronization failure in some circumstances. #5509

Captive Portal

- Fixed captive portal database handling issue that could reset database instead of waiting for lock to clear. #5622
- Fixed problem with 0 byte files in captive portal file manager. #5642

3.3.24 2.2.5 New Features and Changes

Security/Errata Notices

- Updated to FreeBSD 10.1-RELEASE-p24
  - FreeBSD-SA-15:25.ntp Multiple vulnerabilities in NTP [REVISED]
  - FreeBSD-SA-15:14.bsdpatch: Due to insufficient sanitization of the input patch stream, it is possible for a patch file to cause patch(1) to run commands in addition to the desired SCCS or RCS commands.
  - FreeBSD-SA-15:16.openssh: OpenSSH client does not correctly verify DNS SSHFP records when a server offers a certificate. CVE-2014-2653 OpenSSH servers which are configured to allow password authentication using PAM (default) would allow many password attempts.
  - FreeBSD-SA-15:18.bsdpatch: Due to insufficient sanitization of the input patch stream, it is possible for a patch file to cause patch(1) to pass certain ed(1) scripts to the ed(1) editor, which would run commands.
  - FreeBSD-SA-15:20.expat: Multiple integer overflows have been discovered in the XML_GetBuffer() function in the expat library.
  - FreeBSD-SA-15:21.amd64: If the kernel-mode IRET instruction generates an #SS or #NP exception, but the exception handler does not properly ensure that the right GS register base for kernel is reloaded, the userland GS segment may be used in the context of the kernel exception handler.
  - FreeBSD-SA-15:22.openssh: A programming error in the privileged monitor process of the sshd(8) service may allow the username of an already-authenticated user to be overwritten by the unprivileged child process. A use-after-free error in the privileged monitor process of the sshd(8) service may be deterministically triggered by the actions of a compromised unprivileged child process. A use-after-free error in the session multiplexing code in the sshd(8) service may result in unintended termination of the connection.
- pfSense-SA-15_08.webgui: Multiple Stored XSS Vulnerabilities in the pfSense® WebGUI
  - The complete list of affected pages and fields is listed in the linked SA.
- Updated strongSwan to 5.3.3
- Updated PHP to 5.5.30
- Updated miniupnpd to 1.9.20150721 to address a potential vulnerability in miniupnpd.

User Management/Authentication

- Added support for GUI auth from RADIUS to obtain group names from the RADIUS reply attribute "Class" as a string (local groups must exist, similar to LDAP). #935
- Added an LDAP server timeout field to address GUI access issues when the LDAP server is down/unreachable. #3383
- Added support for LDAP RFC 2307 style group membership. #4923
- Worked around a chicken-and-egg problem in user syncing which was preventing users from using ssh the first time the account was saved. #5152
- Prevent deletion of system users and groups by authenticated, authorized users using manually crafted POSTs. #5294

OpenVPN

- Fixed an incorrect netmask being sent to OpenVPN clients with static IP addresses set in RADIUS. #5129
- Changed the calculation of the OpenVPN point-to-point server IP address obtained from RADIUS to be consistent with CSC/Overrides (Server should be one IP address below the Client)

IPsec

- strongSwan upgraded to 5.3.3. strongSwan's change log
- Fixed missing DH group 22-24. #4918
- Fixed handling of IPv4 IPsec Phase 1 endpoints that resolve to an IPv6 address. #4147 (Fixed by strongSwan update to 5.3.3)
- Brought back "auto" IKE version and fixed problems with its previous implementation.
- Pre-shared keys configured as "any" under VPN>IPsec, Pre-Shared Keys tab are added as %any to ipsec.secrets now, as described in the note on the page. #5246
- Resolved memory leak by switching printf hooks to vstr. #5149
- Change to vstr to fix memory leak broke SMP status plugin. Switched to vici for status output.
- ID selectors omitted from ipsec.secrets for mobile PSK+XAuth configurations. Fixes pre-shared key mismatches with Apple iOS Cisco IPsec and other mobile clients. #5245
- Fixed logging default settings and ability to set logging to silent. #5340
- Logging settings applied correctly on clean start and stop/start of service. #5242
- Remove deleted CAs, certificates and CRLs from strongswan configuration. #5238
- Prevent over-matching of auto-added firewall rules for mobile IPsec configurations. #5211
- Added IPv6 virtual address pool support for mobile. #5284
- Allow both IPv4 and IPv6 in phase 2 entries on a single phase 1 when using IKEv2. #5305
- Omit NAT rules for disabled phase 1 and 2 configurations. #5320
- Only display certificate authority field for methods where it's relevant. #5323
- Only write out CA certificates for those specified in a Phase 1 configuration. #5243
- Fixed Hybrid RSA + xauth. #5207

- Fixed configuration of split tunnel attribute. #5327
- Specify rightca in ipsec.conf where relevant. #5241
- Specify leftsendcert=always in ipsec.conf for mobile profiles using IKEv2 to better accommodate iOS and OS X manual configurations. #5353
- Fix IKEv2 mobile client pool status display with small number of active leases

Rules/NAT

- Fixed handling of url_port alias types when processing items that should be handled by filterdns. #4888
- Fixed handling of line endings when parsing a URL table ports file.
- Fixed handling of empty bogon lists on NanoBSD.
- Fixed handling of 6rd rules so they are only added when there is an IPv4 IP defined for the gateway, otherwise the ruleset ends up invalid. #4935
- Added support for port ranges on Outbound NAT. #5156
- Added a check to prevent renaming an alias to an existing name. #5162
- Improved the fix for increasing the "self" table size in pf.
- Imported fixes from FreeBSD for a situation that could result in a panic/crash due to source address limits in pf rules ("pf_hashsrc: unknown address family 0"). #4874

Captive Portal

- Implemented an alternate method to find VIP targets that should be allowed for Captive Portal. #4903
- Improved handling of the captive portal database files for zones in cases when the database files may be corrupt or unreadable. #4904
- Improved handling of vouchers that are too short. In certain cases they were not being properly rejected. #4985
- Fixed handling of voucher database files, initializing the database properly when necessary. #5113
- Fixed loading of allowed hostnames at boot time. #4746, #5345

Packages

- Fixed handling of package install errors and connect timeouts during the install process. #4884
- Improved package version comparison. #4924
- Fixed an issue with package editing where the default value was not being populated for new fields.
- Fixed removal of syslog.conf entries during package uninstall #5210

DHCP

- Fixed handling of DHCP pools that are out of range, preventing them from creating an invalid dhcpd configuration. #4878
- Added support for UEFI network booting with arch 00:09. #5046
- Fixed a situation where dhcpleases could miss updates for hostnames in the leases file, delaying functional hostname resolution of new and updated DHCP leases. #4931
- Automatically add firewall rules to permit DHCP traffic when DHCP Relay is enabled, matching the behavior for DHCP Server. #4558

Interfaces

- Fixed identification of IPv6 interfaces with PPP-type interfaces and DHCP6 #3670
- Removed "Could not find gateway for interface…" log messages as they were largely useless. #4102
- Added OpenVPN interfaces to the list of available interfaces when reassignment is necessary during config.xml restoration.
- Fixed interface assignment menus running off VGA screen.
- Fixed preservation of MLPPP settings when saving interface settings. #4568
- Correct handling of SLAAC, DHCP6 and DHCP-PD with PPP interfaces. #5297

Dynamic DNS

- Fixed Cloudflare support for Dynamic DNS updates.
- Fixed GratisDNS support for hosts without subdomains.
- Disabled DHS provider. It had never worked.
- Fixed IPv4 dynamic DNS registrations on dual stack hosts to providers with AAAA records. #3858
- Update Dynamic DNS using gateway groups upon enable and disable of gateways. #5214
- Fixed Dynamic DNS using gateway groups specifying a CARP IP. #4990

Misc

- Fixed the configuration version comparison in XMLRPC sync to prevent more invalid synchronization cases. #4902
- Cleaned up old unused platforms referenced in a few areas of the code that were no longer relevant.
- Fixed killing of individual states in cases when the source and destination were reversed. #4907
- Fixed killing of individual states for IPv6. #4906
- Changed the "enableallowallwan" script to also allow bogons, which makes the use of RFC 5735 / RFC 6890 test networks easier in lab environments.
- Fixed handling of VIPs in source address selection for Diagnostics > Test Port. #4986
- Updated status.php to include more information. #5304
- Fixed handling of the description in Traffic Shaping.

- Fixed pfSense base version comparison. #4925
- Fixed handling of multiple notices in the same second. #4879
- Removed the routed service as it is being handled by the package.
- Set MIME type for SVG in lighttpd configuration.
- Improved handling of the cron service reconfiguration process.
- Added option to display monitor IP on Gateways widget #4782
- Added "Description" as a display option on Traffic Graphs. #4783
- Fixed handling of L2TP server interface selection. #4830
- Added /usr/bin/dc back into the build. #5111
- Fixed a crash/panic "Sleeping thread owns a non-sleepable lock" in ARP code when using Proxy ARP type VIPs. #4685
- Added support for Sierra Wireless 7355. #4863
- Updated time zones. #5254
- Added fsync of Unbound's root.key to ensure the file isn't corrupted if power is lost shortly after writing of the file. Code added to detect corrupt root.key and delete and recreate it. #5334
- Fix changing outbound NAT modes and uploading/downloading files on exec.php with non-English languages. #5342, #5343
- Associate intermediate internal CA certificates with the signing CA. #5313

3.3.25 2.2.4 New Features and Changes

Security/Errata Notices

- pfSense-SA-15_07.webgui: Multiple Stored XSS Vulnerabilities in the pfSense® WebGUI
  - The complete list of affected pages and fields is listed in the linked SA.
- FreeBSD-SA-15:13.tcp: Resource exhaustion due to sessions stuck in LAST_ACK state. Note this only applies to scenarios where ports listening on pfSense itself (not things passed through via NAT, routing or bridging) are opened to untrusted networks. This doesn't apply to the default configuration.
- Note: FreeBSD-SA-15:13.openssl does not apply to pfSense. pfSense did not include a vulnerable version of OpenSSL, and thus was not vulnerable.
- Further fixes for file corruption in various cases during an unclean shut down (crash, power loss, etc.). #4523
  - Fixed pw in FreeBSD to address passwd/group corruption
  - Fixed config.xml writing to use fsync properly to avoid cases when it could end up empty. #4803
  - Removed the 'sync' option from filesystems for new full installs and full upgrades now that the real fix is in place.
  - Removed softupdates and journaling (AKA SU+J) from NanoBSD, they remain on full installs. #4822
- The forcesync patch for #2401 is still considered harmful to the filesystem and has been kept out. As such, there may be some noticeable slowness with NanoBSD on certain slower disks, especially CF cards and to a lesser extent, SD cards. If this is a problem, the filesystem may be kept read-write on a permanent basis using the option on **Diagnostics > NanoBSD**. With the other above changes, risk is minimal. We advise replacing the affected CF/SD media by a new, faster card as soon as possible. #4814

- Upgraded PHP to 5.5.27 to address CVE-2015-3152 #4832
- Lowered SSH LoginGraceTime from 2 minutes to 30 seconds to mitigate the impact of MaxAuthTries bypass bug. Note *Sshlockout* will lock out offending IP addresses in all past, current and future versions. #4875

Certificates

- Changed the built-in certificate manager to specify keyUsage and extendedKeyUsage in certificates. Windows will now correctly function with IKEv2 using certificates from the built-in certificate manager without disabling EKU. #4580
  - Note: This change applies only to new certificates, created on 2.2.4 or newer, and the CN of the certificate must match the hostname or IP address to which clients connect.
- Added authorityKeyIdentifier to CRLs generated by the built-in certificate manager. (strongSwan requires it to match.) #4860

IPsec

- Fixed non-GCM AES modes with AES-NI enabled. #4791
- Fixed issues with keyid and some mobile IPsec identifiers. #4811 #4806
- Fixed includes so PHP shell session restartipsec script works.
- Fix dashboard hardware crypto display where AES-NI is enabled. #4809
- Fixed issues with IPsec with certificates/ASN1.DN. #4792 #4794
- Added code to write out CRLs from the built-in certificate manager for use by strongSwan.
- Added option for enabling Strict CRL Checking (strictcrlpolicy in strongSwan config).
- Fixed saving Advanced IPsec options before IPsec is enabled.
- Changed LAN bypass to be from “LAN subnet” to “LAN subnet” rather than from “LAN subnet” to “LAN address” to allow it to work for VIPs on the interface.
- Remove “Auto” key exchange option, and change upgraded configurations to IKEv2. #4873
- Specify rightid for mobile IPsec non-PSK configurations. Add peer ID option “any” for excluding peer identifier checks for mobile IPsec circumstances where peer ID matching is impossible or undesirable.

OpenVPN

- Fixed handling of OpenVPN automatic stop/start when bound to gateway groups using CARP VIPs. #4854

DHCP

- Fixed issues with IPv6 Prefix Delegation caused by an invalid prefix/subnet check added to the ISC DHCP daemon. Reported upstream and patched the checks out in FreeBSD ports. #4829

DNS Resolver

- Changed Unbound to use interface-automatic where interface list is empty so it replies correctly in a default HA configuration. #4807
- Fixed selection of a CARP VIP for outgoing interface. #4852
- Fixed some inconsistencies in text across the GUI in places that specified DNS Forwarder vs. Resolver. #4551

Load Balancer

- Improved handling of port ranges in relayd. #4810
- Fixed references to Load Balancer Virtual Server *redirect_mode*.

Traffic Shaping

- Fixed adding of VoIP rules from traffic shaper wizard where IP/alias was not specified. #4838
- Fixed default CoDel values.
- Corrected inverted target/interval values for CoDel.

Rules/NAT/Aliases

- Fixed a foreach() error when saving an empty alias.
- Fixed input validation on Alias import page.
- Fixed inconsistencies in descriptions in Alias editing for URL Table aliases.
- Added labels to more default firewall rules.
- Avoid an error loading the rules with a numeric hostname in an alias.

Misc

- Removed unnecessary deletion of rc.conf; Added an empty rc.conf with a note.
- Removed a check for a QinQ interface existing when deleting. The check unnecessarily made QinQ un-deletable where the parent interface no longer existed.
- Fixed GratisDNS support.
- Fixed glob for serial devices to match more accurately.
- Fixed a foreach() warning when editing PPP entries.
- Fixed GRE and GIF interface input validation so required fields and descriptions match.
- Changed the behavior of Cancel buttons to be consistent (return to referring page).
- Fixed display of advanced DHCP settings when present.
- Removed old, unused *NetUtils.js*.
- Retain */usr/bin/fsync* from FreeBSD in images.
- Added “netstat -ni” to /status.php output.
- Fixed a typo in upgrade code for Captive Portal.

- Fixed limiter upgrade code to allocate pipe numbers even if no rules are present.
- Fixed upgrade code to remove old CA/Cert config entries that were moved/relocated.

3.3.26 2.2.3 New Features and Changes

Security/Errata Notices

- pfSense-SA-15_06.webgui: Multiple XSS Vulnerabilities in the pfSense® WebGUI
  - The complete list of affected pages and fields is very large and all are listed in the linked SA.
- FreeBSD-SA-15:10.openssl: Multiple OpenSSL vulnerabilities (Including Logjam): CVE-2015-1788, CVE-2015-1789, CVE-2015-1790, CVE-2015-1791, CVE-2015-1792, CVE-2015-4000
  - NOTE: pfSense ships with a default set of DH parameters due to the time/CPU they require to generate. A new set of DH parameters may be generated by the user at any time as described in *DH Parameters*
- Fixes for filesystem corruption in various cases during an unclean shut down (crash, power loss, etc.). #4523
  - Changed new filesystems to use the 'sync' option to avoid loss of data.
  - Added upgrade code to activate the 'sync' option on the root slice for existing installations.
  - Changed new filesystems to use softupdates and journaling (AKA SU+J).
  - Changed the way fsck is handled at boot time:
    - Followed best practice of using fsck from FreeBSD rc.d/fsck script. (Run preen mode first and later try forcefully fixing issues.)
    - Added as much information during boot on the status of the filesystem as possible.
    - Changed fsck to run with -C flag and always in foreground during boot to prevent issues that might schedule background mode.
- The forcesync patch for #2401 was considered harmful to the filesystem and removed. As such, there may be some noticeable slowness with NanoBSD on certain slower disks, especially CF cards and to a lesser extent, SD cards. If this is a problem, the filesystem may be kept read-write on a permanent basis using the option on **Diagnostics > NanoBSD**.

Rules/Aliases/NAT

- Fixed a problem with more than 64 IP addresses in the "self" table in pf.
- Fixed issues with FQDNs in aliases causing static entries to be lost. #4296
- Added the tracker ID rule number lookup to dynamic firewall log. #4730
- Fixed alias rename and delete not being propagated to outbound NAT. #4701
- Fixed tracker IDs of policy route negation rules which had been duplicating the tracker ID of the rule they were based upon. This confused the log parser and displayed the negation rule rather than the actual rule. #4651
- Fixed logging of passed IGMP traffic when the rule is not set to log. #4383
- Fixed a situation where a combination of L2TP, overlapping subnets, port forwards and NAT reflection could cause an invalid ruleset. #4772
- Added a GUI field to control the size of the pf fragment limit #4775

IPsec

- Updated strongSwan to 5.3.2. #4750
- Integrated a patch from https://wiki.strongSwan.org/issues/951 to solve IPsec SA rekey issues on strongSwan+FreeBSD. #4686
- Added patches from FreeBSD PR 200282 to help address IPsec rekey issues.
- Backported FreeBSD r283146 and patch from FreeBSD PR 192774 to address PF_KEY ACQUIRE missing port and protocol information.
- Added reply-to/route-to rules for mobile-ipsec. #4235
- Removed the manual specification of reqid in the IPsec configuration because strongSwan 5.3.0 has fixed issues with its handling, which caused the existing code to misbehave. #4665
- Fixed the display and behavior of the LAN bypass option for IPsec. #4655
- Fixed IPsec LAN bypass toggling every time save is pressed. #4640
- Changed how charon is started and restarted to fix a various issues with IPsec configuration reloading. #4268
- Added new modes for IPsec Phase 1 according to RFC 5903 (Ecliptic Curve groups). #4260
- Implemented the "make before break" feature available in strongSwan 5.3.0, which is useful for IKEv2. #4626
- Fixed vpn_ipsec_configure so it always performs a filter reload to ensure the ruleset is updated where necessary in every IPsec change scenario. #4631
- Added support for EAP-RADIUS to IKEv2 Mobile Clients. #4614
- Fixed a panic/crash when accessing services on the firewall over mobile IPsec on 32-bit installations (set net.inet.ipsec.directdispatch=0 on i386). #4537
- Fixed an issue with FQDN hosts and PSKs. #4785

OpenVPN

- Added a space to the OpenVPN TLS Verify script to avoid appended parameters appearing the same as existing parameters.
- Fixed get_interface_ip() to return the IP address correctly for gateway groups specifying a VIP, which fixed OpenVPN clients not working with gateway groups specifying VIPs. #4661
- Changed the OpenVPN client settings to allow just one of either the username or password to be specified. #3633
- Fixed OpenVPN servers listening on an associated IPv6 addresses.

Captive Portal

- Fixed filterdns to use the proper API for ipfw changes on FreeBSD 10.1+ to correct captive portal allowed hostnames not being loaded into tables at boot time. #4746
- Fixed Captive Portal RADIUS accounting. #4131
- Fixed Captive Portal Idle-Timeout causing a value of 2147483647 for acctsessiontime. #4652
- Fixed disconnection of active voucher users, and corrected disconnection of users especially when triggered via XMLRPC. #4625

Operating System

- Fixed both the kernel and choparp to better handle I/O and prevent issues in the way it handles BPF, which can contribute to a panic when using Proxy ARP VIPs. #4685
- Merged a patch that avoids a panic on sockbuf module. #4689
- Fixed AESNI to be SMP friendly to avoid various decryption errors and possible encryption mistakes. Also present critical_enter/critical_exit to avoid preemption of the currentrunning thread which should fix panics. #4702
- Updated time zone data from FreeBSD 10.1-RELEASE. #4459
- Fixed creation of */var/spool/lock* on NanoBSD at boot time. #4532
- Removed boot_serial='yes' from loader.conf when serial is disabled. #4617
- Fixed an issue where mtree would fail during an upgrade from a previous version of FreeBSD when moving to 2.2.x. #4653

Interfaces/NIC Drivers

- Added support for Sierra Wireless MC7354.
- Added support for Intel X552, ixgbe changes from stable/10, and moved altq changes for ixgbe to the large ixgbe patch.
- Enabled ix/ixv/ixl modules in the kernel
- Fixed duplication of statistics on vlan(4) interfaces for outgoing bytes #3314
- Fixed updating wireless statistics so that the output bytes are not always zero. #4028
- Added a patch from FreeBSD PR 200722 for mpd5 to preventing it from printing a warning when renaming an interface to an existing name.
- Fixed SLAAC/DHCPv6 handling for cases where the global SLAAC IPv6 address might be present when using DHCPv6. #4483
- Corrected descriptions on Key Rotation and Master Key Regeneration for wireless interfaces.
- Removed the "insert my MAC" feature from interfaces.php.
- Defined $var_path as a global key since it is being used in interfaces.inc, but it was not declared.
- Fixed issues setting the MTU on certain interfaces. #4397

Packages

- Fixed various issues with PBI generation.
- Synchronized and cleaned up various pfPorts, eliminated several that had changes pushed back into FreeBSD ports.
- Fixed an issue where rebuild_package_binaries_pbi.php could fail due to missing build files. #4600
- Backported patches from FreeBSD stable/10 to fix a crash when stopping squid. #4592
- Fixed pfflowd to use the correct version for parsing the new pfsync header and corrected the pfsync version check. #4304
- Updated pkg_edit.php with fixes for usecolspan2 and combinedfields.

- Fixed pagination on pkg.php.
- Fixed boot-time log file initialization for package logs. #4603

DHCP/RA

- Clarified that DNS Forwarder and Resolver both apply in DHCP/DHCPv6 and router advertisements. #3730
- Removed unnecessary filtering on the DHCP static mappings table.
- Added appropriate RA Flags for "Stateless DHCP".
- Added error checking to avoid warnings about DHCP relay during boot.
- Fixed hostname validation for static DHCP leases such that only fully qualified hostnames must be unique, not only short names.
- Fixed adding DHCP static mappings from the DHCP leases view to non-default pools. #4649
- Stopped invalid DHCP settings from being applied when input errors exist.
- Removed DHCP static lease overlap cleanup and its associated function and killing of the DHCP daemon. This behavior could cause problems with failover scenarios, especially when adding/editing/removing static mappings.

Web GUI

- Fixed language selection. #4705
- Changes to status.php to make it easier to gather and submit support information:
  - Added sanitization of OpenVPN static/tls keys to status.php.
  - Cleaned up, organized, and expanded the info presented by status.php.
  - Changed status.php to additionally save the output to individual text files and compress them into a .tgz for later download.
- Fixed setup wizard LAN DHCP pool calculation to avoid an invalid pool.
- Improved the setup wizard hostname check. #4712
- Fixed some minor text issues in wizards.
- Changed the wizard to use the current WAN gateway name rather than assuming the name. #4713
- Updated and corrected the wireless status flags and capabilities list. There are many more possible flags, now documented at *Wireless Status*.
- Added a fall back to look up local user privileges and groups if the groups could not be found from LDAP and there is a local user.
- Fixed Crash Reporter submissions when symlinks were present as part of crash report, which would fail to save the report on the server. #4650
- Set a user agent for the Crash Reporter.
- Cleaned up code logic in status_upnp.php.

CARP

- Changed CARP so that it does not trigger a carp demotion taskqueue if the value is 0, which can cause the cluster to misbehave.
- Fixed issues for CARP+Bridges where pfSense would crash or freeze. #4607
- Fixed the CARP plugin call for packages. The "interface" parameter was coming through as NULL during CARP events.
- Added INIT event for CARP in devd.conf as an alternate for 'backup', otherwise scripts would not take down services during a MASTER->INIT transition. (e.g. interface unplug, link loss)
- Fixed NTP so that it properly uses selected CARP IP addresses. #4370
- Fixed CARP packet flow after initial interface creation. #4633

Traffic Shaper/Limiters

- Fixed limiters when used with IPv6. #2526
- Corrected handling of NAT when RDR/BINAT is applied on packet and it is being sent to limiters. #4596

DNS

- Consistently handle clear_subsystem_dirty after an Unbound restart.
- Added a call to clear_subsystem_dirty('staticmaps') when using Unbound, otherwise DHCP static mappings would not fully apply when Unbound was in use. #4678
- Fixed an Unbound warning when "dnsallowoverride" was off and port forwarding was on. #4682
- Re-enabled verification for selfhost DynDNS since their chain issue has been resolved. #4545

Misc

- Updated PHP to 5.5.26
- Fixed various issues in the installer for GEOM mirrors (mirror slice detection, gmirror cleanup on non-clean disks.) #4658
- Fixed new user creation to use skel as the source of new user files rather than copying from the home directory of *root*.
- Changed growl so it will not be called if the configured address isn't an IP address or resolvable hostname. This avoids 1 minute timeout delay in fsockopen in growl.class. This change cuts that down to about a 20 second timeout. #4739
- Added a reboot after restoring a full backup in the GUI. #4107
- Deprecated */usr/local/bin/3gstat* as it was no longer used. It was replaced by *3gstats.php* long ago.
- Started using the "host!" flag when setting CURLOPT_INTERFACE, as recommended by the CURL documentation.
- Started passing the interface to CURLOPT_INTERFACE instead of the IP address, also started using the "if!" flag to avoid CURL trying to resolve the interface name.
- Fixed NTP serial configuration to setup the serial port before attempting to configure a GPS unit.
- Cleaned up various HTML/XHTML issues.

- Fixed a check for deleting a VIP when in use by OpenVPN.
- Fixed issues with backup/restore of a config.xml breaking the serial console on ADI installs. #4720
- Fixed several issues with boot speed when WAN was disconnected. #4442
  - Reduce the timeout for HTTP/HTTPS connection attempts for items like URL table aliases. Once connected, they can run past that. 5 seconds should be more than enough for any properly-functioning network.
- Removed some unused/obsolete files.

3.3.27 2.2.2 New Features and Changes

Security/Errata Notices

- pfSense-SA-15_05.webgui: Multiple XSS Vulnerabilities in the pfSense® WebGUI
- FreeBSD-SA-15:09.ipv6: Denial of Service with IPv6 Router Advertisements. Where a system is using DHCPv6 WAN type, devices on the same broadcast domain as that WAN can send crafted packets causing the system to lose IPv6 Internet connectivity.
- FreeBSD-SA-15:06.openssl: Multiple OpenSSL vulnerabilities. Most aren't applicable, and worst impact is denial of service.

Rules / NAT

- Added hidden config option to disable blocking of link-local IPv4 (169.254.0.0/16) for the rare instances where it's required. Not recommended, violates RFC 3927.
- Fixed invalid ruleset generation when using port forwards with destination "any" on a DHCP client WAN-type interface, have pure NAT mode reflection enabled, and have the interface with link up but unable to reach a DHCP server for an extended period. #4564
- Allow the use of version IPv4+IPv6 on firewall rules without restrictions on protocol. The former restrictions date back to earlier base software versions, and are no longer applicable.
- Omit route-to from rules specifying a specific gateway when that gateway is forced down. #4566
- Use the subnet address when forming rules for networks, rather than the interface IP address
- Added SCTP to the protocol drop-down for firewall rules

IPsec

- Enforce disabling of "prefer old SAs" option. When the GUI configuration checkbox was removed in 2.2.1, it fell through to the default of the underlying software in many cases, leaving the option enabled instead of disabled. Having this option enabled will cause connectivity problems after rekeying in many circumstances. Upgrading to 2.2.2 will fix this.
- strongSwan upgraded to 5.3.0
- Don't apply mobile IPsec phase 2 PFS configuration to non-mobile IPsec. #4538
- Correct applying of uniqueid configuration. #4359
- Bring back automatic exclusion of LAN subnet to LAN IP for scenarios where remote IPsec overlaps with local LAN subnet. #4504
- Enable ike_name for daemon logging, adding connection identifiers to IPsec logs that can be correlated to output of 'ipsec statusall' (GUI log viewer integration to come).

DNS Forwarder/Resolver

- Fix DNS registration of hostname "0" #4573
- Domain overrides to multiple server IPs are possible in DNS Resolver. Add message noting this, and how to achieve it. #4350
- Always configure user-specified DNS servers in the Unbound configuration, to make its behavior consistent with dnsmasq
- Only list nameservers once in resolv.conf

Wireless

- Atheros wireless driver updated to latest from FreeBSD 11-CURRENT. Not many changes since 2.2.1-RELEASE. #4582
- Wireless cards removed from ALTQ-capable interfaces (traffic shaper capability) since that isn't supported at the moment. #4406
- New option "auto" added for Standard. This omits configuring mode with ifconfig, which currently can trigger driver problems that don't exist when not specified. Standard "auto" is preferred, and possibly required, for BSS and IBSS wireless modes with Atheros cards (at a minimum, potentially others).

IPv6

- Make sure 'DHCPv6 Prefix Delegation size' is provided if 'Send IPv6 prefix hint' flag is checked to avoid generating invalid dhcp6c configuration file.
- DHCPv6 Relay fixed. #4572
- Allow "0" for id-assoc na ID, id-assoc pd ID, sla-id and sla-len DHCP6 configuration options. #4547
- Fix the use of multiple prefixes in IPv6 router advertisements. #4468

Other

- Clean up logic in OpenVPN resync code. Discussion here and additional change here.
- SSL certificate validation disabled for selfhost - their certificate chain had a problem that made OpenSSL fail verification, making the service non-functional. #4545 The provider fixed the issue after 2.2.2-RELEASE, so verification has been re-enabled for 2.2.3 and newer.
- Fix error in traffic shaping wizard. #4529
- Fix broken image path. #4530
- A variety of minor text clean up in web interface.
- Remove some code no longer used in a few places.
- Clean up of code path when adding a new user. #4620
- Make sure RRD backup is not restored when /var memory disk is not in use. #4531
- Show friendly name of the interface on custom RRD graph drop-down selection
- PHP upgraded to 5.5.23
- Prevent a user from adding a VLAN using the invalid ID "0"

- Cleanup display of times in DHCP leases
- Use the correct field for voucher "expired" and "no access" messages
- Fix traffic shaper wizard bandwidth input validation calculations [https://redmine.pfsense.org/issues/4259 #4259
- Changed Diagnostics > Sockets to display sockets bound to localhost
- Allow single interface bridges, useful for span ports and when migrating interfaces to a bridge

3.3.28 2.2.1 New Features and Changes

Security/Errata Notices

- pfSense-SA-15_02.igmp: Integer overflow in IGMP protocol (FreeBSD-SA-15:04.igmp)
- pfSense-SA-15_03.webgui: Multiple XSS Vulnerabilities in the pfSense® WebGUI
- pfSense-SA-15_04.webgui: Arbitrary file deletion vulnerability in the pfSense WebGUI
- FreeBSD-EN-15:01.vt: vt(4) crash with improper ioctl parameters
- FreeBSD-EN-15:02.openssl.asc: Update to include reliability fixes from OpenSSL

Potentially Relevant

The following updates are included from upstream in FreeBSD, but are not directly relevant. Neither pfSense software nor its packages include SCTP services, but such services may have been manually added by the user.

- FreeBSD-SA-15:02.kmem: SCTP SCTP_SS_VALUE kernel memory corruption and disclosure
- FreeBSD-SA-15:03.sctp: SCTP stream reset vulnerability

Not Relevant

- OpenSSL "FREAK" vulnerability:
  - Does not affect the web server configuration on the firewall as it does not have export ciphers enabled.
  - pfSense 2.2 already included OpenSSL 1.0.1k which addressed the client-side vulnerability.
  - If packages include a web server or similar component, such as a proxy, an improper user configuration may be affected. Consult the package documentation or forum for details.

Known Issues

- Some cases remain where filterdns does not properly handle hostnames in multiple aliases properly. Most of the cases have been fixed, so the situation is better than 2.2-RELEASE, but it is not 100% resolved. See issue #4296 for details. Placing hostname aliases into a separate alias so they are not mixed with static entries effectively works around the issue.

General

- Updated the default SSL cipher list to be stronger, obsoletes the need for a "BEAST protection" option #4230
- Fixed gen_subnet_max returning an incorrect result on 32 bit (i386) versions, which in turn fixed Wake on LAN and other areas on 32 bit (i386) versions. #4318
- Fixed crash on boot with some hardware, caused by gpioapu on systems where smbios.system.product is null. Mostly seemed to be the recycled Watchguard users affected by this issue. #4363
- Updated ufslabels.sh to handle a wider variety of disk layouts.
- Added a choice of SMTP authentication protocols for notifications, Office365 mail support. #4176
- Removed latin-1 encoding of RSS feed to fix display issues of RSS items.
- Fixed an issue where the GUI setting for PAP or CHAP in L2TP Server was not being respected.
- Fixed changing source tracking value separate from changing the Sticky option.
- Added input validation to force a minimum *100000* byte log file size to prevent undersizing the logs.
- Added more cleanup to the **Restart PHP-FPM** console menu action.
- Removed PTR records for aliases in host overrides.
- Fixed *diag_arp.php* to allow underscore in resolved host names.
- Fixed an issue in DHCP settings where the "add routers" value was not being preserved across a loop for each interface.
- Added capability to handle reverse lookup domain overrides.
- Fixed issues with NTP RRD graph state changes.
- Added input validation to require RADIUS protocol and server IP address/host in Captive Portal when RADIUS authentication is selected. #4384
- Fixed swap size calculation in the installer to avoid creating improperly sized partitions in systems with lots of RAM but not much disk space.
- Fixed test for comconsole when matching for enabling serial console. #4464
- Updated pfSense PHP shell help to current configuration structure. #4492
- Fixed switching from a PPP type WAN to "None" or "DHCP".
- Disables SNMP hostres module on APU boards until we figure out why it's crashing on this specific board. #4403
- Removed -U from mtree call used to restore files permissions as it was breaking symlinks on upgrade. #4328
- Added input validation for Wireless configurations to prevent problematic combinations of settings. #4178
- Improved handling of FQDN entries in aliases with filterdns, but not 100% resolved. #4296
- Fixed various typo, style, and formatting issues.

Rules / NAT

- Fixed ordering of DHCPv6 client and bogon rules so the bogon rules can't block DHCPv6 requests. #3395
- Fixed a bug where applying NAT changes in Hyper-V could break the running NAT configuration. #4445
- Fixed a bug where marking a packet with only a number resulted in a broken rule. #4274
- Fixed DSCP choices that were non-functional and resulted in a broken ruleset. #4302
- Fixed PHP memory exhaustion on NAT pages with VIP ranges on a 32 bit (i386) versions. #4317 (Related to #4318)
- Fixed input validation on Outbound NAT to accept a port range. #4300
- Removed Carrier-Grade NAT subnet from "Block private networks" as it was in 2.0.x and earlier releases since it specifically notes RFC 1918 and CGN is more closely related to bogon networks. #4379
- Removed code that set adaptive.start and end to 0, now they are left at their defaults (60% and 120% of the state limit, respectively) if not user-overridden.
- Added configuration options for state timeout values under System>Advanced, Firewall/NAT. #4509

IPsec

- Added MOBIKE control, now disabled by default. #3979
- Fixed page rendering so MOBIKE is only shown with IKEv2 selected, NAT-T only shown with IKEv1 selected.
- Removed *Prefer older IPsec SAs* option from the GUI, and existing configurations with it enabled will not have that setting applied. This is almost never desirable, and with the change to strongSwan it frequently was the source of problems. The very few who might desire such an option can configure the *net.key.preferred_oldsa* sysctl accordingly under **System > Advanced**, **System Tunables**.
- Fixed Phase 2 duplication issue. #4349
- Added input validation to prevent use of AES key lengths larger than 128-bit when the *glxsb* cryptographic accelerator is enabled. #4361
- Added an option for an IPsec tunnel to act as a responder only. #4360
- Added a filter reload when IPsec is disabled. #4245
- Fixed RSA cert handling in IPsec to use double quotes on asn1dn specification so it is properly interpreted by strongSwan. #4275
- Added an option to allow controlling unique ID handling in IPsec advanced settings. #4359
- Fixed *restartipsec* command line script.
- Fixed handling of IPsec with Gateway Groups #4482
- Added a workaround to disable the strongSwan Unity plugin. #4178
- Added error logging when an IPsec Phase 1 cannot be located.

OpenVPN

- Added encoding for username and password to avoid issues with special characters. #4340
- Fixed issues with OpenVPN TLS and authentication scripts. #4329
- Fixed issues with handling of the Authentication Mode if the user changes the value after changing other incompatible settings.

DNS Resolver

- Upgraded to Unbound 1.5.3.
- Added correct scaling of *rrset-cache-size* in *unbound.conf*. #4367
- Added support for 0x20 DNS random bit. #4205
- Changed DNS Resolver default values to be a bit more strict: Enable Hide Identity, Hide Version, Harden DNSSEC data.
- Force harden glue configuration option, and remove GUI control of that option. Problem with Unbound pre-1.5.2 means in 2.2-RELEASE, having this option enabled, and DNSSEC disabled, could lead to DNS cache poisoning. #4402
- Added a check to test if Unbound is enabled and using the same port before allowing dnsmasq to be enabled. #4332
- Removed hard-coded value for *harden-referral-path*. It defaults to *no*, so no behavior change, and that setting is unlikely to ever become a default. This allows users to configure an override to enable this option if desired. #4399

Logging

- Fixed GUI log parser handling for IGMP log entries. #4343
- Fixed syslogd issues where the daemon stopped and failed to restart during boot in some cases. #4393

Traffic Shaping

- Fixed input validation errors in the Traffic Shaper wizard due to old data not being cleared. #4333
- Fixed handling of Upstream SIP Server in the Traffic Shaper wizard. #4314, #4427
- Fixed crash when using limiters and pfsync. #4310
- Fixed limiters used with IPv6. #2526

IPv6

- Fixed calculation of the 6rd default gateway honoring netmasks other than /32.
- Fixed recording of the IPv6 interface's new IP address and do not issue commands that cannot succeed. #3669
- Fixed not being able to save custom and custom-v6 DynDNS entries.
- Added IPv6 IP addresses to */etc/hosts* in the same manner IPv4 IP addresses are added. #4395
- Fix computation of the displayed DHCPv6 range start to be consistent with the actual check.
- Added *dhcp6.name-servers* option with DHCPD-PD regardless of PD length.

- Fixed Net_IPv6::compress() to properly handle all-zeros address.
- Enabled `UnicastOnly` in radvd for `ovpnX` interfaces. #4455
- Removed requesting a prefix delegation when there are no tracking interfaces setup to use it. #4436
- Added code to destroy stf interface when a 6rd or 6to4 tunnel is disabled. #4471

VIP/CARP

- Added input validation to prevent the VIP "interfaces" from being assigned since they are just an identification of the VIP for tracking and not actual interfaces. #4389
- Fixed functions to properly return the VIP subnet now that the CARP might not match its parent interface subnet. #4390
- Fixed a bug that caused the status icon from previous CARP VIP to be shown in cases where the IP address was not present on an interface.
- Changed the carp demotion factors slightly to avoid CARP transitions that are most likely unnecessary. (Do not demote on NIC send errors or pfsync errors)
- Expanded the CARP demotion error
- Added button to reset demotion status
- Fixed handling of IP Alias deletion from a secondary node using XMLRPC configuration sync #4446

Misc Binary/OS Changes

- Upgraded PHP to 5.5.22.
- Re-enabled Suhosin in PHP.
- Updated 802.11 code and Atheros wireless driver from FreeBSD 11-CURRENT
- Added patch to fix crash with Ralink wireless cards in access point mode. #4117
- Added athstats, cryptostats and cryptodev back. #4239
- Fixed AESNI module checks when used inside a virtual machine.

3.3.29 2.2 New Features and Changes

Special Notes

Due to CSS and JavaScript changes, forcing the browser to clear its cache or reload the pages after an update is advised. This is especially true if any cosmetic anomalies are observed, such as alignment problems or spurious bits of text in widgets.

Security Fixes

- Update to openssl 1.0.1k to address FreeBSD SA-15:01
- Multiple XSS vulnerabilities in web interface pfSense-SA-15_01
- OpenVPN update for CVE-2014-8104
- NTP update FreeBSD-SA-14:31.ntp though these circumstances don't seem to impact pfSense® software.

Default Configuration Changes

- **DNS Resolver (unbound) enabled for new installs**. #3396
- **DNS Forwarder (dnsmasq) disabled for new installs**. #3396
- **Change default NICs from vr to em** – vr is on the way out and em is the most common NIC in use today.
- Default config.xml has been cleaned up. Outdated comments have been removed that used to loosely document the config file, but had been neglected for quite some time and aren't all that useful anyway.
- Default sysctls have moved out of config.xml and now reside in globals.inc to reduce the size of config.xml
- Default sysctl values do not need to be set in config.xml. The default values are obtained from sysctl now. Also to reduce config.xml size.
- Tracking IDs added to default rules

Security Enhancements

- Verify SSL certificates for HTTPS URLs
- Detect if an unofficial package repository is in use and warn the user. Warning is displayed on the dashboard and package management pages. #484
- Check and verify the package server's SSL certificate if using HTTPS. #484
- For dyndns providers that support HTTPS, use it when performing updates.
- Replaced lots of GET actions with POST actions in various places in the GUI as they were touched.
- Update jquery to 1.11.1
- Remove almost all calls to history.back() and make Cancel button back to HTTP_REFERER
- Hide FreeBSD version from sshd banner. #3840
- Disable SSLv3 in lighttpd
- Disable RC4 ciphers in lighttpd
- Teach the certificate generation code how to make a self-signed certificate, and change the GUI cert generation code to use it. Also, move the GUI cert generation code to its own function so we can add a GUI option to regenerate it later. Also use some more sane defaults for the contents of the default self-signed certificate's fields so it will be more unique and less likely to trigger problems in browser certificate storage handling.
- Add command line script to generate and activate a new GUI certificate (generateguicert)
- Catch some more sensitive information when sanitizing the contents of config.xml output on /status.php.

OS Changes

- Updated base OS to FreeBSD 10.1-RELEASE
- PHP backend switched from FastCGI to PHP-FPM
- PHP Moved to 5.5
- Migrate captive portal code to SQLite3 PHP module
- Fix some lingering call-time pass-by-reference instances that fail on PHP 5.5
- Default serial speed is now 115200 #3715
- Sync gettytab and etc/ttys with FreeBSD 10-STABLE and reduce customizations
- Log pfSense version to syslog after bootup
- Set the sysctl net.inet.icmp.reply_from_interface to 1 to use the incoming interface to send ICMP replies. #3666
- Switched the hash method in pf to XXHASH for speed improvements

DNS

- Imported Unbound for use as the default DNS Resolver. The old dnsmasq DNS Forwarder is available as a non-default option. Upgraded systems will retain existing settings.
- Various changes to Unbound and supporting programs to complete its integration.
- Removal of bind from FreeBSD base necessitated the switch to alternate programs for DNS utilities (e.g. drill for dig, different nsupdate)
- AJAX DNS updates for firewall logs (when clicked)
- Make sure that the DNS Forwarder/Resolver is always capable of accepting queries on localhost before using it as a DNS server.
- If localhost is configured to be included in resolv.conf, force its selection in Unbound. The resolv.conf logic prevents that from being a problem, but users don't seem to realize they have to pick that to use Unbound for the host itself.
- IPv6 support in Unbound
- Check port of dnsmasq/unbound and skip 127.0.0.1 in resolv.conf if not port 53. #4022
- Add a note to the wizard about the DNS Resolver ignoring manual name servers by default. (They are still used as secondary/tertiary servers for the firewall itself, however)
- Domain and search should not both be defined in resolv.conf per FreeBSD man page and handbook (only the latter is actually used). Only search is set now.

CARP

- Changes to CARP for new FreeBSD 10 CARP system
- Provide a way to 'permanently' set CARP to 'maintenance mode' (advskew 254) persisting through a reboot so the primary machines stays as backup/inactive unless the secondary no longer exists on the network. This can be desirable when there are problems (possibly with the hardware) and the primary machine needs to be booted and checked again before regaining master status.
- Key off net.inet.carp.demotion and display a warning to the user if the system has self-demoted its CARP status.
- Allow CARP IP address to be outside interface and alias subnets

Interfaces

- Implement an option to allow using the IPv4 connectivity interface for sending the dhcpv6 information. Usually useful for PPP[oE] type links and some ISPs
- Add gre and gif checks for for IPv4 function interface_has_gateway($friendly), like they are already for IPv6
- Do not allow the user to set IPs for GRE interfaces on interface edit page. #3575
- On interfaces_assign.php, let user select network port to add instead of picking the first available #3846
- When changing an existing VIP, use previous configured interface for checking, this fixes the issue that happens when trying to change a VIP to a new interface. #3807
- Validate the GIF interface MTU (must be something between 1280 and 8192) #3927
- Properly set MTU for lagg(4) interface #3922
- Fix formatting of the Interfaces Widget on the Dashboard. #3937
- Don't allow interface descriptions that are strictly numbers as that generates an invalid ruleset. #4005
- Disable delete_old_states in dhclient-script. rc.newwanip handles this correctly in 2.2, and this killed states in multiple circumstances where that isn't necessary nor desirable.
- Do not unset configuration values from PPP config if not needed. #3727
- Overhaul handling of flags for hardware offloading and make it work correctly for system_advanced page settings. Lagg is still a special case that may require a reboot initially to apply. #1047
- Don't try to launch 3gstats unless it's on a valid device.
- Updated list of mobile service providers

Gateways/Routing

- Add an option to force a gateway to be down. #2847
- List GWGs in Interface to send DynDNS update from
- Allow reordering, batch delete, and disable of static routes
- Option to disable a gateway added
- Check gateway for IPv6 also for reply-to rules.
- Fix issue where ICMP6 messages sometimes have the wrong source IP address when a monitor IP address has been set #3607
- Improve look of gateways widget
- Provide a toggle for apinger debug messages to be logged to syslog
- Setting an interface IP to 0.0.0.0 with mask 0.0.0.0 overwrites the default route with that interface's link route. Follow FreeBSD 10.1 and use a /8 mask instead. #3941
- Use static route with -iface option for PPPoE to help when more than one PPPoE connection has the same gateway. #4040
- net.inet6.ip6.rfc6204w3 needs to be 1 for dhcpv6 to work correctly. #3361
- Add a route debug option to log info about route commands executed (where those aren't already logged) to help with troubleshooting various routing scenarios.
- Make sure srcip and target have scope when link-local addresses are used in apinger. #3969

- Properly generate and use the default gw for 6rd.

Firewall Rules

- Custom logging daemon that provides easy-to-parse output on a single line
- Persistent tracking ID for firewall rules so that logs may always be traced back to their corresponding rules
- Removed settings for maximum tables and maximum table entries since pf on FreeBSD 10 does not have any limits for these.
- Expose all p0f OS types that it supports so that subtypes of various Operating Systems can be detected (e.g. blocking Windows XP)
- The "(self)" concept of "Any IP address on this firewall" is now a choice for firewall rule destination (and floating rule source for out direction rules), port forward destination, and outbound NAT source.
- Can now optionally log default pass rules as well as default block rules
- Add IP alias subnets to interface subnet macro on GUI. #983
- Adjust states summary for new pfctl -ss output. #2121
- Add a more obvious note on group rules about how they do not work as expected for WANs
- block IPv4 link-local/APIPA 169.254.0.0/16. Per RFC 3927, hosts "MUST NOT send the packet to any router for forwarding", and "any network device receiving such a packet MUST NOT forward it". FreeBSD won't route it (route-to can override in some circumstances), so it can't be in use as a real network anywhere with the possible exception of local-only networks. Unlikely any such situation exists anywhere #2073
- Fix JavaScript confirmation dialog for EasyRule.
- Use 'clog -f /var/log/filter.log' to view firewall log entries from the console so they are displayed in the new format.
- Set MSS clamping on VPNs in both directions rather than requiring it be set on both ends.
- Add option to kill all states on IP change, currently a hidden option for more testing. #1629
- Kill states associated with the old WAN IP when WAN IP has changed. #1629

NAT

- Hybrid outbound NAT style that allows the user to keep the existing automatic behavior but layer manual rules on top of it.
- Option to disable outbound NAT without disabling pf
- Display networks used in automatic outbound NAT when using that mode
- Allow reordering, batch delete, and disable of 1:1 NAT rules
- Take virtual IPs into consideration for automatic outbound NAT rules #983
- Outbound NAT can apply to any type of interface, make WAN-type specific reference generic

Aliases

- Allow individual line descriptions on alias bulk import
- Implement URL Table aliases for ports
- Optimizations for URL table aliases to use less memory and be more robust in general
- Alias name cannot have more than 31 chars, add maxlength to the field as an extra check. #3827
- Prevent Internal Server Error if an IP range is entered backwards.
- Expand range or subnet entered into a host type alias.
- Warn that IPv6 address ranges are not supported in aliases.
- When an alias contain hosts, add IPs and networks to filterdns too, otherwise the ruleset ends up with a predefined and non-persistent table. #3939

Dashboard & General GUI

- Various fixes for XHTML compliance
- Various fixes for typos
- Add a setting to allow the user to specify the clog file size so more (or less) entries may be kept in the raw logs. Retain previous default size values if the user has not specified a preferred size. Files can only be resized when initialized, so provide a "Reset All Logs" button as well to force clear all logs and set them up at the new size.
- Add an option for users to be able to adjust how many configuration revisions are kept in the local backup cache.
- Show backup file size in config history.
- Display pfSense interface name on status interfaces
- Dashboard cleanups/fixes for jQuery
- Add "pfsense_ng_fs" full screen/widescreen theme
- GUI redirect works on both IPv4 and IPv6 #3437
- Disk usage section of the System Information widget now shows all UFS, ZFS, and cd9660 filesystems, not just the root (/) slice, and also indicates if they are a RAM disk.
- Add a message about premium content to the setup wizard and add a link in the menu to the signup page.
- Add pages missing from the Status > Traffic Graph privilege that are required for the full page to load
- Fix traffic graph widget default autoscale
- Be more strict on user and group removal to avoid removing accidentally removing additional users #3856
- Add an option to restart php-fpm from console
- Add .inc file for gmirror status widget to give it a better title and link to the management page.
- Allow the Virtual IP list table to be sorted (cosmetic only)

Translations

- Change default charset on pages to utf-8
- Updates to pt_BR translation
- Added Japanese translation
- Added Turkish translation
- Fixes for gettext

Captive Portal

- Add a way to download CP portal, error and logout html pages. #3339
- Add an option to restore default logout/error/portal custom pages on Captive Portal. #3362
- For more than 100 MAC pass-through entries create pipes in line with the rules file to speedup the process. #3932
- Zone backend changed from text-based (e.g. "cpzone") to using the zone id (e.g. "2") for specifying the context.
- ipfw_context has been removed. To list zones, use "ipfw zone list"
- Default lighttpd daemon port for a Captive Portal zone is based on the zone ID. For example, zone ID 2 uses port 8002. There may not be a daemon on port 8000.

IPsec

- IPsec backend changed from racoon to strongSwan
- IKEv2 settings have been enabled in the GUI
- Default IPsec configuration settings for newly created site to site configurations updated to use main mode and AES 256 on both phase 1 and 2.
- IPsec status page and dashboard widget changes to accommodate different output from strongSwan
- Move the IPsec settings from System > Advanced, Misc tab to "Advanced Settings" tab under VPN > IPsec.
- It is now possible to configure *L2TP/IPsec*
- Add AES-GCM and AES-XCBC to the list of available IPsec algorithms and hashes, respectively. Expand P1 DH groups up to 24.
- Allow hash algorithms to be empty for phase 2 where the encryption is AES-GCM
- Allow to reorder IPsec Phase 1 and Phase 2 items, remove multiple P1/P2 items, toggle enable/disable status of P1/P2 items #3328
- Provide a first implementation of EAP-TLS authentication with IKEv2. It is a start and might not work on all cases
- Do not accept non-ASCII characters on IPsec PSK #3931
- Fix ping_hosts.sh to not ping IPsec if CARP is in backup.
- Allow accept_unencrypted_mainmode_messages to be enabled for IPsec if needed.
- Check that subnet masks are equal when choosing binat type for IPsec to avoid errors on ruleset. #3198
- Change NAT Traversal options as strongSwan only has two options: force or auto.

- Don't allow P2 local+remote network combinations that overlap with interface+remote-gateway of the P1. #3812

OpenVPN

- Allow entering OpenVPN client credentials in the GUI
- Add fields for local (push route) and remote (iroute) network definitions in an OpenVPN client-specific override entry.
- Change OpenVPN compression settings to cover the full range of allowed settings on OpenVPN (unset, off, on, adaptive) rather than a simple off/on switch that either doesn't set the value or enables it with adaptive (OpenVPN's default).
- Add an Authentication Digest Algorithm drop-down to OpenVPN server/client and to the wizard (SHA1 is the default since that is OpenVPN's default)
- Add option to specify client management port for OpenVPN client export use
- Ensure e-mail address carries over from the CA screen to the Cert screen in the OpenVPN wizard.
- Allow the user to select "None" for OpenVPN client certificate, so long as they supply an auth user/pass. #3633
- Byte counts on OpenVPN status are now human readable rather than huge unformatted numbers.
- OpenVPN instances have new options: "Disable IPv6", route-nopull, route-noexec, verb selector
- Use stronger defaults in the OpenVPN wizard.
- Fix ovpn-linkup for tun + topology subnet case setting router as ifconfig_local envvar when route_vpn_gateway and ifconfig_remote are both not defined. #3968

DHCP

- Add code for UEFI booting and DHCP
- Advanced RFC 2136 configuration for DHCPd service
- Add ability to not supply a DHCP gateway to clients
- Allow defining DHCP static mappings using dhcp-client-identifier
- Do not call write_config() when Applying Changes on DHCP settings #3797

Packages

- Package signing to ensure validity/authenticity
- Single package manifest (XML) file rather than one per architecture
- Various improvements to PBI setup/structure from upstream (PC BSD)
- Added the capability for package hooks in /etc/rc.carpmaster and /etc/rc.carpbackup
- Split package category display into separate tabs for categories, and provide an "All" tab
- Move the fetching of a package's config file and additional files to separate functions, and then have the "xml" package button perform these so that it is not only a redundant copy of the "pkg" reinstall button. This can help ensure a package files are in a known-good state before other actions are performed, in case the deinstall would fail or behave erratically due to other files being missing.

- Clarify logs generated by newwanip(v6) when restarting packages, it's not only IP changes that end up here (by design).
- When reinstalling a package, try to start it after the install completes.

Dynamic DNS

- Added support for DynDNS Provider "City Network"
- Added support for DynDNS Provider "OVH DynHOST"
- Added support for DynDNS Provider "GratisDNS"
- Added support for DynDNS Provider "Euro DNS"
- Added support for DynDNS Provider "CloudFlare"
- Add support for custom IPv6 DDNS.
- Add backend support for HE.net AAAA record updates.
- Add additional options to Custom DynDNS
- Allow hostname to start with '@.' for namecheap #3568
- Do not disable certificate verification in DynDNS. Proper CA certificates are now in place to validate SSL in these cases.
- "+" is a valid character in some dynamic DNS providers' usernames. #3912

GEOM Mirrors (gmirror)

- New gmirror library to perform various gmirror tasks and get information, using some of the former widget logic to start.
- Added a Diag > GEOM Mirrors page that displays information about existing mirrors and performs various management tasks. This will only show in the menu if a gmirror is detected at bootup. Current actions include rebuilding a drive, forgetting disconnected mirror drives, insert/remove, deactivate/activate, clearing metadata. It's now possible to use the GUI to rebuild a failed mirror by performing a forget, then insert action to replace a missing/dead drive.
- Also included is a notification setup. Mirror status is polled every 60 seconds, and if any aspect of the mirror changes, notifications are issued that alert in the GUI and by SMTP, etc.

NOTE: If a manual gmirror configuration was performed post-install and not using the pfSense installer gmirror option before install, there is a chance that the mirror will not function on pfSense 2.2 because the manual post-install method did not create a completely proper mirror setup. If the upgraded mirror does not function on 2.2, the following `/boot/loader.conf.local` entry may be used to work around the integrity check that would otherwise fail:

```
kern.geom.part.check_integrity=0
```

If one of these configurations is present, we strongly recommend backing up the configuration and reinstalling using the built-in gmirror option in the pfSense installer.

Traffic Shaping

- Fix DSCP values and provide a config upgrade to fix values stored in config.xml. #3688
- Remove 'multi lan/single wan' and 'multi wan/single lan' traffic shaper wizards, multi lan/wan can be used to replace any of them.
- Only show the correct type of interfaces (LAN/WAN) on traffic shaper wizards #3535
- Shaper wizard will automatically attempt to guess the correct number of WANs and LANs.
- Updated and expanded traffic shaping for games, game consoles, and other applications.
- Allow up to 2900 limiters. This was set to 30. #3213
- Fix logic to find available next number for limiters and queues. #3998
- Add vmx and hn to list of ALTQ capable interfaces.
- Remove the "Limiter burst" parameter as it currently doesn't work with dummynet in pf.

Misc

- Cleaned up various older files/scripts that were no longer being used
- Dropped all support for cvsup. cvs is dead, long live svn and git.
- Optimizations/changes to the XML Parsing code
- NTP updates to handle a wider ranges of GPS devices and more NTP options
- Move to zerocopy_enable for bpf to optimize logging which uses bpf interface. This should increase the general performance since pflog is always enabled.
- Add sshd service to list (if enabled)
- Add a "status" subcommand to the svc php shell script.
- When using the reset webConfigurator password option on the console, if authentication server is not Local Database, ask user if they want to revert back to it. #3341
- Fix interface selections on UPnP to show the customized descriptions entered by the user. While here, add an external interface selection knob. Fixes #3141
- Layer 7 Pattern: EAOrigin.pat
- Layer 7 Pattern: SWF (Flash)
- Remove some old obsolete code that referred to the now-defunct "embedded" platform that was replaced with NanoBSD back in 1.2.x.
- Sometimes fsck requires a second run, teach rc script to call it more than once when it's necessary
- Add column for internal port on UPnP status page
- Make listening on interface rather than IP optional for UPnP
- Use interface name for miniupnpd rather than IPv4 address #3874
- Packet Capture: Host field supports rudimentary boolean logic. Captures can specify multiple IP addresses and use and/or between IP addresses. Example: To perform an "and" match where both hosts must match: `192.168.1.1, 192.168.1.2`. To perform an "or" match where any of the specified hosts can match: `192.168.1.1|192.168.1.2|192.168.1.3`
- Packet Capture: Protocol, host, and port now support negation.

- Added interface column to Diagnostics > States
- Change is_port() to only validate a single port, we have is_portrange() for specific cases. #3857
- Fix guess_interface_from_ip() to account for differences in netstat output. #3853
- Fix Certificate Authority SAN name handling #3347
- Add a basic command line password reset script.
- Use configured proxy URL/port for downloading bogon list. Does not use credentials. #3789
- Underscores are valid characters in domains. #3219
- Let user decide to proceed with upgrade when sha256 fails to download. #3576
- Remove the command number shown in the shell prompt.
- Use a better method of finding disks for SMART.
- Process obsolete files in shell script instead of PHP.
- Do not allow FQDN in fields that should only accept a hostname.
- Set proxy environment variables on interactive shell and also on crontab so that they may be used by all scripts. #3789
- Add input checkboxes to remove multiple users and groups
- Make sure an empty group or user is not created when editing
- Update URLs in help.php.
- Change wording at the end of the wizard to remove "donate" since that is no longer an option.
- Put the booting signal in globals.inc since it makes all the other scripts detect we are booting. Otherwise separate PHP instances will not detect that. rc.bootup clears this flag so all should work correctly
- Force serial console when it was selected by the installer. #4009
- Wait 10 minutes before retrying bogon fetch on soft failures to avoid us getting DoSed if something is wrong there (like someone's system can't validate the cert)
- Use IPv4 for ntpq if IPv6 is not allowed

HEADS UP for Xen Users

The FreeBSD 10.1 base used by pfSense 2.2 includes PVHVM drivers for Xen in the kernel. This can cause Xen to automatically change the disk and network device names during an upgrade to pfSense 2.2, which the hypervisor should not do but does anyway.

The disk change can be worked around by running /usr/local/sbin/ufslabels.sh *before* the upgrade to convert the fstab to UFS labels rather than disk device names.

The NIC device change issue has no workaround. Manual reassignment is required at this time. Note there have been performance issues reported in Xen with this NIC device change.

3.3.30 2.1.5 New Features and Changes

The pfSense® software version 2.1.5 release follows shortly after *2.1.4* and is primarily a security release.

Security Fixes

- pfSense-SA-14_14.openssl
  - See http://www.openssl.org/news/secadv_20140806.txt
  - Updated to OpenSSL 0.9.8zb and 1.0.1i
- pfSense-SA-14_15.webgui
- pfSense-SA-14_16.webgui
- pfSense-SA-14_17.webgui

Other Fixes

- Handle a missing DHCPD config section properly during a configuration upgrade
- Fix a regression that broke CARP+IP alias VIP functionality
- Fix the Pass, Block, Reject and Interface filters in the Firewall Logs Widget #3725
- Use HTTPS for dyndns providers that support it
- Avoid resetting the firewall hostname from a WAN DHCP server #3746
- Add missing qlimit keyword in some shaper rules
- Change Cancel button to call history.back() when editing firewall aliases to fix issues with IE 11 #3728
- Allow hostnames in bulk import since they are valid entries in a network type alias
- Fix input validation logic on diag_testport.php, escape more shell arguments for good measure
- Escape the individual dnsmasq advanced/custom options
- Encode the detail field of an alias entry before displaying its contents back to the user
- Encode interface/VIP descriptions before displaying them on the NTP daemon settings, and GIF/GRE interfaces
- Per the dhcpd.conf man page and other documentation from ISC, mclt must not be defined on the secondary
- Shorten the wait at "reload" in startup wizard to 5 seconds from 60
- Do not execute DNS lookups on GET, only pre-fill Host box so the user can press the button to execute
- Turn alias creation links from DNS lookups into submit buttons for POST
- Remove javascript alert DNS resolution action from the firewall log view. It was already removed from 2.2, and it's better not to allow a GET action to perform that action
- Require click-through POST confirmation when restoring or deleting a configuration from the backup history page
- Avoid a "Cannot use string offset as an array" error if the packages section of the config is missing
- Avoid generating an invalid IPsec (racoon) config if the user specified a mobile pool that is too small
- IPsec phase 2 pinghost was not used if the source IP was a virtual IP address #3798
- Move dhcp6c log to dhcpd.log #3799

- Do not reset source and destination port range values when it's an associated rule created by NAT port forward. #3778
- Added filter.so to list of extensions loaded for filter_var() support.
- The pfSense PHP module was setting the subnet mask of lo0 to /0, which could break some routes and cause other unintended routing side effects.

3.3.31 2.1.4 New Features and Changes

pfSense® software version 2.1.4 follows very shortly after *2.1.3* and is primarily a security release. Refer to the *2.1.1 release notes*, *2.1.2 release notes*, and *2.1.3 release notes* for other recent changes.

Security Fixes

- pfSense-SA-14_07.openssl
  - FreeBSD-SA-14:14.openssl
- pfSense-SA-14_08.webgui
- pfSense-SA-14_09.webgui
- pfSense-SA-14_10.webgui
- pfSense-SA-14_11.webgui
- pfSense-SA-14_12.webgui
- pfSense-SA-14_13.packages

Packages also had their own independent fixes and need updating. During the firmware update process the packages will be reinstalled properly. Otherwise, uninstall and then reinstall packages to ensure that the latest version of the binaries is in use.

Other Fixes

- Patch for Captive Portal pipeno leaking issue which leads to the 'Maximum login reached' on Captive Portal. #3062
- Remove text not relevant to Allowed IPs on the Captive Portal. #3594
- Remove units from burst as it is always specified in bytes. (Per ipfw(8)).
- Add column for internal port on UPnP status page.
- Make listening on interface rather than IP optional for UPnP.
- Fix highlighting of selected rules. #3646
- Add guiconfig to widgets not including it. #3498
- /etc/version_kernel and /etc/version_base no longer exist, use php_uname to get the version for XMLRPC check instead.
- Fix variable typo. #3669
- Delete all IP Aliases when an interface is disabled. #3650
- Properly handle RRD archive rename during upgrade and squelch errors if it fails.
- Convert protocol ssl:// to https:// when creating HTTP headers for XMLRPC.

- Show disabled interfaces when they were already part of an interface group. This avoids showing a random interface instead and letting the user add it by mistake. #3680
- The client-config-dir directive for OpenVPN is also useful when using OpenVPN's internal DHCP while bridging, so add it in that case also.
- Use curl instead of fetch to download update files. #3691
- Escape variable before passing to shell from stop_service().
- Add some protection to parameters that come through _GET in service management.
- Escape argument on call to is_process_running, also remove some unnecessary mwexec() calls.
- Do not allow interface group name to be bigger than 15 chars. #3208
- Be more precise to match members of a bridge interface, it should fix #3637
- Do not expire already disabled users, it fixes #3644
- Validate starttime and stoptime format on firewall_schedule_edit.php
- Be more careful with host parameter on diag_dns.php and make sure it's escaped when call shell functions
- Escape parameters passed to shell_exec() in diag_smart.php and elsewhere
- Make sure variables are escaped/sanitized on status_rrd_graph_img.php
- Replace exec calls to run rm by unlink_if_exists() on status_rrd_graph_img.php
- Replace all `hostname` calls by php_uname('n') on status_rrd_graph_img.php
- Replace all `date` calls by strftime() on status_rrd_graph_img.php
- Add $_gb to collect possibly garbage from exec return on status_rrd_graph_img.php
- Avoid directory traversal in pkg_edit.php when reading package xml files, also check if file exists before try to read it
- Remove id=0 from miniupnpd menu and shortcut
- Remove . and / from pkg name to avoid directory traversal in pkg_mgr_install.php
- Fix core dump on viewing invalid package log
- Avoid directory traversal on system_firmware_restorefullbackup.php
- Re-generate session ID on a successful login to avoid session fixation
- Protect rssfeed parameters with htmlspecialchars() in rss.widget.php
- Protect servicestatusfilter parameter with htmlspecialchars() in services_status.widget.php
- Always set httponly attribute on cookies
- Set 'Disable webConfigurator login autocomplete' as on by default for new installs
- Simplify logic, add some protection to user input parameters on log.widget.php
- Make sure single quotes are encoded and avoid javascript injection on exec.php
- Add missing NAT protocols on firewall_nat_edit.php
- Remove extra data after space in DSCP and fix pf rule syntax. #3688
- Only include a scheduled rule if it is strictly before the end time. #3558

3.3.32 2.1.3 New Features and Changes

pfSense® software version 2.1.3 follows very shortly after *2.1.2* and is primarily a security release. Refer to the *2.1.1 release notes* for changes from 2.1 to 2.1.1 and *2.1.2 release notes* for changes from 2.1.1 to 2.1.2.

Security Fixes

- pfSense-SA-14_05.tcp
  - FreeBSD-SA-14:08.tcp
- pfSense-SA-14_06.openssl
  - FreeBSD-SA-14:09.openssl

Packages also had their own independent fixes and need updating. During the firmware update process the packages will be reinstalled properly. Otherwise, uninstall and then reinstall packages to ensure that the latest version of the binaries is in use.

Although these security issues warrant updating as soon as possible, they are of relatively minor impact to the average user. According to the FreeBSD SA, the TCP flaw is mitigated by scrub in pf which is enabled by default in pfSense. The OpenSSL flaw is not used by any daemons in the pfSense base system and only certain packages make use of the affected feature, so the impact there is also minimal.

Other Fixes

- Various fixes to accommodate recent changes/optimizations in the tools repository
- Move clog binary to its proper place in /usr/local/ to respect hier(7)
- Fix remove button on Diagnostics > Tables #3627
- Fix more potential places for interface looping in OpenVPN and with normal interfaces
- Fixes for URL table alias updates (locking, reload)
- Fix IPsec Phase 1 duplication
- Fix 'add rule on top of the list' allowing after param to be -1
- Correct Captive Portal redirection URL to unbreak ones passed through Radius attributes and respect user choices.
- Make miniupnpd listen on interface instead of IP
- Don't refuse to delete a bridge in the GUI just because its bridge interface doesn't exist, just log that it doesn't exist and don't attempt to ifconfig destroy it, delete it from config
- Fixes for DynDNS to allow configurable check host.
- Resolver has no option for remote syslog, remove wrong copy/paste that was adding it when apinger was enabled
- Fix typo for GIF tunnels to work over IPv6
- Fix for dhcrelay target using default GW
- List Gateway Groups in Interface to send update from for custom DynDNS entries

3.3.33 2.1.2 New Features and Changes

pfSense® software version 2.1.2 follows very shortly after *2.1.1* and is primarily a security release. Refer to the *2.1.1 release notes* for changes from 2.1 to 2.1.1.

Security Fixes

The Heartbleed OpenSSL bug and another OpenSSL bug were both covered by the following security announcements:

- pfSense-SA-14_04.openssl
  - FreeBSD-SA-14:06.openssl
  - CVE-2014-0160 (Heartbleed)
  - CVE-2014-0076 (ECDSA Flaw)

Packages also had their own independent fixes and need updating. During the firmware update process the packages will be reinstalled properly. Otherwise, uninstall and then reinstall packages to ensure that the latest version of the binaries is in use.

Other Fixes

- On packages that use row_helper, when user clicks on add or delete button the page scrolls to top. #3569
- Correct typo on function name in Captive Portal bandwidth allocation
- Make extra sure that we do not start multiple instances of dhcpleases if, for example, the PID is stale/invalid and there is still a running instance.
- Fix CRL editing. Use an alphanumeric test rather than purely is_numericint because the ID is generated by uniqid and is not purely numeric. #3591

3.3.34 2.1.1 New Features and Changes

Security Fixes

- FreeBSD-SA-14:01.bsnmpd / CVE-2014-1452
- FreeBSD-SA-14:02.ntpd / CVE-2013-5211
- FreeBSD-SA-14:03.openssl / CVE-2013-4353, CVE-2013-6449, CVE-2013-6450

  Note: This FreeBSD SA is only for the FreeBSD 10.x base, but we include that version of OpenSSL from ports.
- The following FreeBSD Security Advisories were not relevant to pfSense® software:
  - FreeBSD-SA-13:14.openssh
  - FreeBSD-SA-14:04.bind
- Use HTTPS to get updates. #2952
- Escape necessary chars to avoid XSS injection. #2952
- Add escapeshellarg() calls on more exec parameters.
- Replace some exec() calls by php functions like symlink, copy, unlink, etc.
- Use HTTPS for pfsense.org URLs.

- Protect output to browser by using htmlspecialchars. #3461
- Improve checks for params 'id', 'dup' and other similar ones to make sure they are numeric integer, also, pass them through htmlspecialchars before printing.
- Remove special characters that can lead to shell/XSS compromises from submitted input when installing packages. #3461
- Ask for validation when real package operation will be done and ask for the operation with POST to get protection from CSRF. #3460
- Use HTTPS for fetching packages.

Interfaces

- Updated em/igb/ixgb/ixgbe drivers that add support for i210 and i354 NICs and fix issues with `ix(4)` cards.
- Prevent assigned vlans from having their tag changed.
- Fix ifconfig error on gif in certain cases.
- If rc.newwanip is run on an interface that should not have an IP address, do not take any action. This could lead to certain interfaces bouncing link if they had no IP address.
- In rc.newwanip, if the interface is configured and not enabled, bail. We do not need to change settings for disabled interfaces. #3313
- Skip processing in rc.newwanip if the interface has no IP address.
- Fix pkg_edit.php to show interface description instead of interface name
- Make sure vlan interface exist when they are configured #3270
- Limit CIDR choices for IPv4 on GRE interface. #3277
- Do not destroy an interface when it's being disabled #3350
- Prevent network or broadcast address to be set on interface (console, GUI and wizard). #3196
- Reduce unnecessary operations and other fixes to MTU code. This fixes slow boot times and proper handling of mtu for VLANs.
- Provide a dynamic gateway for GIF and GRE v6 tunnels so it can be used on firewall rules etc. #3484
- Bring up appropriate interface for GRE/GIF. #3281
- Prevent removing the IP from the underlying GRE interface in the OS when assigning GRE interface and configuring an IP address. #3280
- When an interface goes down try to shut the RAs and dhcpd6 service on that interface. #2627
- Sync up ALTQ-capable interfaces list
- Trigger rc.newwaipv6 from pppoe when it gets an inet6 configuration
- Update list of mobile service providers.
- Correct check to enable ieee8021x.

Gateways/Routing

- Respect default gateway option when adding a gateway from interfaces page. #3230
- Use a more accurate error message when attempting to add/edit a gateway that does not have an appropriate IP address for the type. #3282
- Make return_gateways_array() return all disabled gateways when $disabled is true. #3291
- Don't flush interface cache on each call of the function when looping through all gateways.
- Fix an issue that changes wrong gateway entry when items are hidden
- Delete static route when monitor IP is removed, also save monitor IP even when it's disabled
- Return Gateway Group IP protocol version even when no gateway IP can be located.
- Remove broken 'dynamic6' gateway, we already have ipprotocol to tell us the IP version, leave it more simple using only 'dynamic'

NAT/Firewall Rules/Aliases

- Reload filter rules when activate or deactivate dhcpdv6 #3218
- Make sure no extra spaces end up in the parsed IP in the filter logs as it can lead to issues in other places (Easy Rule, etc)
- Use (self) rather than any as the destination for the lockout rules
- Use (self) instead of any for web lockout
- Avoid pf table names conflict. #3268
- Fix display of full URL in URL table listing as seen in an Alias popup. #3242
- Make it more explicit that 'update freq.' for URL table aliases unit is days
- Fix situation where removing an alias entry and then adding a new one resulted in an entry box with broken formatting. #3283
- Make sure pf rule labels never have more than 63 chars. #3208
- Rewrite the display_host_results() function to use spaces instead of tabs. It does a much better job of aligning the fields in each column and works in all the browsers, particularly chrome which doesn't support the tab character.
- Handle comma-separated list of remote networks when making vpn_networks table
- Fix rules that pass out traffic for Proxy ARP VIP entries which had incorrect destination #3331
- Load only the options rather than clearing the whole ruleset.
- Validate IP address ranges correctly on Alias Bulk Import
- Fix display of CIDR/Update Freq in Alias Edit
- In the filter log, the protocol might also say "icmpv6" so account for that when making a rule using Easy Rule.
- Move 'allow dhcpv6 client' rules above block bogonsv6 ones. #3395
- Only add dhcpv6 client allow rules if ipv6allow is set
- Add all advanced options to rule table hover text.
- Open up Firewall Rules Advanced Options section if any values have been set.
- Validate rule Advanced Options numeric entries properly

- Disable default allow incoming rules for 6to4 and 6rd interfaces. This rule unintentionally allows all services on the interface.
- Skip OpenVPN interfaces when creating the first set of manual rules to be consistent with the behavior of Automatic Outbound NAT. #3528
- Try to restore last working ruleset rather than staying without configuration at all if an invalid ruleset is encountered.
- Fix days and weeks selection on schedules
- Prevent prevent putting an subnet in the IPv6 address field since it then breaks the filter generation process.
- Put a timeout of 30 seconds on the bogon update download. #3412
- Before downloading file to process urltable, there is a random wait time between 5 and 60 seconds. Because of this, the difference between file mtime and current time can be less than $freq * 86400 and it'll be skipped. Add 90 seconds (60 of max random wait + 30 just to be sure) to avoid skipping a file that should be updated. #3469
- Validate if src OR dst have IP address set when protocol is IPv4+v6. #3499
- Improve data validation to avoid save a host/subnet or a IPv4 with invalid mask. The reported error is on javascript and only happen on IE8, but this fix will prevent the same issue happening in the future on a different browser. #3449

Traffic Shaping

- Fixed typo in CoDel wiki link
- Fix codel not being applied on non-priq queue types
- Fix saving and range checking of 'Packet loss rate' and 'Bucket Size' in limiters.
- Add previously missing DSCP VA.
- Clarify note on limiter queue weight to state that higher values get a larger share.

Dashboard & General GUI

- Convert mac address to lowercase when saving to avoid duplicates. It fixes #3195
- Include the CP zone in the form parameters if one is defined. Fixes access to concurrent graph on zones other than the first/default.
- Miscellaneous HTML cleanup
- Fix interface names shown in the traffic graphs widget. #3245
- Send the help links to HTTPS destinations on web servers that support HTTPS.
- Specify favicon in pages directly
- Add some missing privileges to the list. #3279
- Many fixes on privileges. #3216
- Allow setting a default scale type preference for the traffic graphs widget
- Account for a widget being null/not defined, and not just closed/open when deciding if a widget function should be called. This allows the system information dashboard widgets to update properly.
- Avoid dashboard divide by zero errors
- Detect Zones and Cores for thermal sensors using regex. #3337

- Do not sort users when adding privileges. It's unnecessary and lead to unintentional edits to the wrong account.
- Add specific privilege for easyrule.
- Return all stats when all or remote is selected on Traffic Graph and make the default query return "Local" traffic.
- Update year, links for 2.1.1.

Captive Portal

- Fix CP stats generation for concurrent users. #3225
- Remove redundant copies of getNasIP() #3234
- Set default captive portal RADIUS authentication value to radius_protocol during upgrade #3226
- Add Captive Portal Zones privileges definition. #3216
- Prevent a possible division by zero in Captive Portal. #3212
- Fix saving of voucher sync settings
- Reduce the total minutes by the remote minutes used, do not use the value directly. Otherwise the voucher will be cut short or listed invalid when it otherwise should have time left over.
- Make sure to give the Captive Portal zone a name during the upgrade, or else it comes through with a blank/null name.
- Properly set zone dedicated rules in the rules/pipes DBs to properly release when a zone is deactivated
- Don't generate rules for disabled captive portal instances
- Do some more error checking and put secondary radius attributes only if configured on a Captive Portal instance.
- If set use the default bandwidth setting on the Captive Portal even for MAC passthrough.
- Fix various problems with Captive Portal voucher synchronization introduced during conversion to zones.
- Properly compile the Captive Portal database query to insert the values.
- Fix deletion of IPFW rules and pipes for passthru MAC. #3538
- Use the 11th column for the radius context rather than overriding the interim interval field with it. #3447
- Use descr as the field name for voucher description so it gets CDATA protection. #3441
- Consider setting of noconcurrent login for passthrough expiration of users. #3340
- Use the default bandwidth specification if configured even for allowed IP address and hostname.
- Properly detect when there are issues with communicating with syncip and to use the local DB for this. Otherwise detect if the remote says the voucher is not valid say its not valid.

VPN

- Fix find_service_by_openvpn_vpnid() on OpenVPN Status
- Allow special characters to be used on IPsec mobile login banner. #3247
- Fix cisco-avpair processing for IPsec and OpenVPN, and route processing from avpair replies.
- Fix logic in detecting if OpenVPN resync needed
- Fix vpn_pppoe_get_id and stop duplicating pppoeid for multiple servers. #2286
- Use env var provided by openvpn to determine if it's tun or tap. #3475

- Add an option to verify IPsec peers_identifier when it's ASN.1 distinguished name. #2904

Certificates

- Certificate Manager, for 'Create an internal Certificate' use the correct 'Digest Algorithm'
- OpenSSL does not like country codes longer than two letters, so remove entries that are not actually country codes.
- Perform a much more accurate comparison between two certificates to determine if they are identical when checking their revocation status. #3237
- Allow an "empty" CRL to be exported, since this is still a valid action.
- Fixes for "Alternative Names" on certificates.
- Fix issue with CSR generation. #2820
- Increase default openssl to bits 2048.

DHCP

- Optimize DHCPv4 lease display online status for static leases. Do not re-parse complete ARP table for each lease, as it can be slow with large ARP tables.
- Add upgrade code to change the DHCP next-server value to nextserver since it was renamed sometime in 2.1 but upgrade code didn't follow.
- Give clients the IPV6 address of the DNS server via DHCPv6 Server
- Check if dhcp start and end addresses are inside interface subnet. #3196
- Remove 'deny unknown clients' option from DHCPv6 since it's not supported. #3364
- Fix DHCP lease time display, strftime already convert it to local timezone, so we no need to calc offset
- Use correct parameter (bootfile-url) to configure netboot on DHCPdv6. #3421
- Only use IPv4 DNS servers in IPv4 DHCP configuration. #3483
- Fix PHP error when saving DHCP settings if no manually configured DNS servers exist.
- Send a HUP to dhcp6 to signal a reload. #3514

Load Balancing

- Prevent a Fall Back Pool from being selected when the DNS protocol is in use. If one is present in the config, ignore it. #3300
- Fix display of pools in the LB status widget and on the LB Virtual Server status.

Time

- Allow multiple valid time servers to be entered in the wizard, as they are allowed under System > General
- Update time zone data to 2013i
- Teach system_timezone_configure() to deal with symlinks to avoid having timezone misconfigured. #3293
- Add 'limited' to ntpd restrict list to workaround FreeBSD-SA-14:02.ntpd/CVE-2013-5211. #3384
- Use "disable monitor" in NTP config to mitigate FreeBSD-SA-14:02.ntpd/CVE-2013-5211.
- Update ntp to ntp-devel for FreeBSD-SA-14:02.ntpd/CVE-2013-5211.
- Avoid placing an empty "interface listen" directive in ntpd.conf.

Misc

- Fix ALIX upgrade crash during RRD processing
- Fix "Could not open shared memory for read 1000" issue on Diagnostics > NanoBSD. #3235
- Fix ufslabels.sh logic to avoid trying to convert slices which are already using appropriate labels. Fixes #3207
- Fix removal of the first cron job entry in the list.
- We do not use nor include newsyslog, so remove the cron job from the default configuration and on upgrade.
- Split SSL/TLS into separate checkboxes so that plaintext connections can be made secured by using STARTTLS. Support for SMTPS connections should probably be done away with in future. #3180
- Add source address selection to syslog settings, so it can work more effectively over a VPN. #355
- Rework the usage of the shell i/o during stop_packages(), fixes the "Syntax error: bad fd number" for the remaining people who still saw it on shutdown
- Switch to rw mode before file operations on RFC2136 cache. Fixes #3201
- Make the RADIUS settings respect the description of the timeout field. If the timeout value is left blank, use 5 seconds, don't print an error.
- Call conf_mount_rw before deleting a user. #3294
- Handle the reinstallall case with confirmation. #3548
- Do not list the same CARP ip as an option for its own Interface.
- Accept adding an IP Aliases on top of CARP VIP when the parent interface does have a valid IP address in the alias subnet.
- Simplify log filtering logic calling grep less times, as done on mail_reports.inc on 2c6efc9.
- Fix console recent config restore, allow restoration of the last backup listed. #3438
- Enhanced validation of general DNS servers and gateways
- Add a mechanism by which the serial port can be forced on always regardless of the config setting. (useful for nano+vga setups)
- Add a knob to let the user select which console (video or serial) is preferred in cases where there are multiple consoles present.
- Skip input validation when choosing an existing certificate in the User Manager. #3505
- pfSense_interface_deladdress() only knows how to delete an ip address, not a subnet. #3513
- Make is_linklocal case-insensitive. #3433

- Errors in in RRD graph calculations
- Delete /var/crash content when the user clicks 'No'. #3486
- Make sure filesystem is read-write when operating on groups. #3492
- Fix OpenVPN XML section name for selective configuration backup.
- Remove TRIM_set and TRIM_unset support. This method isn't very elegant and isn't necessary in the long run. It's better handled during the install process or while booted off other media (e.g. CD or Memstick).

3.3.35 2.1.0 New Features and Changes

Security Fixes

Three FreeBSD security advisories are applicable to prior pfSense® software releases. These aren't remotely exploitable in and of themselves, but anyone who can execute arbitrary code on the firewall could use one or more of these to escalate privileges. FreeBSD-SA-13:13.nullfs FreeBSD-SA-13:12.ifioctl.asc FreeBSD-SA-13:09.ip_multicast.asc

IPv6 Support

IPv6 Added to many areas of the GUI. At least the following areas/features are IPv6-enabled. Others may work as well

- Aliases (Firewall) - Aliases can contain both IPv4 and IPv6, only addresses relevant to a given rule will be used
- CARP RA
- CARP Failover
- DHCP Server w/Prefix Delegation
- SLAAC WAN
- 6to4 WAN
- 6to4 WAN w/Prefix Delegation
- 6rd WAN
- 6rd WAN w/Prefix Delegation
- DHCP6 WAN
- DHCP6 WAN w/Prefix Delegation
- DHCPv6 Relay
- DNS Forwarder
- Firewall Rules
- Gateway Groups/Multi-WAN - See *Configuring Multi-WAN for IPv6*
- Gateway Status (apinger)
- GIF Tunnels
- GRE Tunnels
- GUI Access
- IPsec
- L2TP

- NPt
- NTP
- OpenVPN
- Packet Capture
- PPPoE WAN
- Router Advertisements
- Routing
- Server LB
- Static IP
- Syslog (remote)
- Limiters (dummynet pipes)
- Virtual IPs - IP Alias
- Virtual IPs - CARP
- DNS from RA
- Accept RA when forwarding
- Auth via RADIUS
- Auth via LDAP
- XMLRPC Sync
- RRD Graphs
- DHCP Static Mapping - Works by DUID
- DynDNS (HE.net hosted DNS, RFC2136, custom)
- MAC OUI database lookup support for NDP and DHCPv6. (Was already present for DHCP leases and ARP table) requires the nmap package to be installed to activate

NOTE: Unlike earlier snapshots, BETA, etc, currently we do NOT flip the "Allow IPv6" checkbox on upgrade, to preserve existing behavior. To activate IPv6 traffic, a user will have to flip this setting manually

Packages

- PBI (push button installer) package support - all of a package's files and dependencies are kept in an isolated location so packages cannot interfere with one another in the way that was possible on 2.0.x and before using tbz packages
- RIP (routed) moved to a package
- OLSRD moved to a package
- Unbound moved back to a package (Will try integration again for 2.2)
- Increase the verboseness of the package reinstallation process in the system logs for a post-firmware-update package reinstallation operation

OS/Binary/Supporting Program Updates

- Based on FreeBSD 8.3
- Updated Atheros drivers
- OpenSSL 1.0.1e (or later) used by OpenVPN, PHP, IPsec, etc
- PHP to 5.3.x
- OpenVPN to 2.3.x
- Added mps kernel module
- Added ahci kernel module
- Updated ixgbe driver
- Many other supporting packages have been updated

Dashboard & General GUI

- Switch from Prototype to jQuery
- Improved navigation and service status in the GUI (shortcut icons in each section to quickly access config, logs, status, control services, etc)
- Multiple language support, a mostly-complete translation for Brazilian Portuguese is included
- Read-only privilege to create a user that cannot modify config.xml
- Dashboard update check can be disabled
- Fixed theme inconsistencies between the login form and other parts of the GUI
- Various fixes to pages to reduce potential exposure to certain CSRF/XSS vectors
- Updated CSRF Magic
- Set CSRF Magic token timeout to be the same as the login expiration
- Added IE Mobile for WP8 to list of browsers that get an alternate theme at login
- Truncate service status so long package descriptions cannot break formatting of the status table
- Many fixes to HTML/XHTML to improve rendering and validation
- Added a note to the setup wizard letting the user know that it can be canceled at any time by clicking the logo image
- Make dashboard update check respect nanobsd-vga #3078
- Firewall Logs Widget filtering and column changes
- Added totals for some dashboard widget meters (memory, swap, disk usage)
- Changed dashboard display for states and mbufs to be meters, and to show usage as a percentage
- Update dashboard mbuf count via AJAX
- Show a count and layout of CPUs in the dashboard if multiple CPUs are detected

Captive Portal

- Multi instance Captive Portal
- Multiple Captive Portal RADIUS authentication sources (e.g. one for users, one for cards)
- Logic fixes for voucher encryption
- Many optimizations to Captive Portal processing, including a database backend and moving functions to a php module to improve speed
- Optional Captive Portal user privilege
- Add checks to make sure CP hard timeout is less than or equal DHCP server default lease time, to avoid issues with CP sessions being valid for incorrect IPs, and users switching IPs while they should still be connected to the portal
- Fixes for captive portal voucher syncing on HTTPS with a custom port #3001
- Fixes for custom Captive Portal files leaving symlinks on the filesystem after files were removed
- Added MAC OUI database lookup support to CP status (requires nmap package to be installed)

OS/System Management

- Ability to select serial port speed
- Added a manual way to enable TRIM if someone needs it
- Added a manual way to trigger a fsck on reboot
- AES-NI support (Cryptographic Accelerator feature on new Intel/AMD CPUs) – Still experimental, not supported by some areas of the OS yet.
- Support for certain thermal sensors via ACPI, coretemp, and amdtemp
- System startup beep can be disabled
- Separate powerd setting for when on battery
- Add optional ability to change the size of RAM disks for /var/ and /tmp/ for systems that have RAM to spare
- Add optional ability for full installs to use RAM disks for /var/ and /tmp/ as is done on NanoBSD. Reduces overall writes to the media, should be more SSD-friendly
- Use a custom sysDescr for snmp similar to m0n0wall's format. Fixes #2893
- Added tunable to allow disabling net.inet.udp.checksum - disabling UDP checksums can improve performance, but can also have negative side effects
- Added an mtree database with the correct default permissions, owner, sha256 sum, and some other information that is used to verify file permissions post-install and post-upgrade
- APC is not started for PHP unless the system has over 512MB RAM, to reduce memory usage on systems with low RAM

Multi-WAN

- DynDNS multi-WAN failover
- IPsec multi-WAN failover
- OpenVPN multi-WAN failover
- Changed descriptions of the values for gateway monitoring
- Display apinger (gateway monitoring daemon) as a service when it is enabled
- Fixes for apinger to reload via SIGHUP properly, to avoid unnecessary restarts and loss of gateway status data
- "State Killing on Gateway Failure" now kills ALL states when a gateway has been detected as down, not just states on the failing WAN. This is done because otherwise the LAN-side states were not killed before, and thus some connections would be in limbo, especially SIP.
- Due to the change in its behavior, "State Killing on Gateway Failure" is now disabled by default in new configurations and is disabled during upgrade. If the feature is desired, it must be manually re-enabled post-upgrade.

NTP

- NTP daemon now has GPS support

IPsec

- More IPsec hash algorithms and DH key groups added, "base" negotiation mode added
- Mobile IPsec supports separate "split dns" field and doesn't just assume the default domain for split DNS domains
- Properly ignore disabled IPsec phase 2 entries
- NAT before IPsec (1:1 or many:1) outbound
- Set default Proposal Check setting to Obey for mobile IPsec
- LDAP and RADIUS are now possible authentication sources for IPsec mobile xauth
- Delete the SPDs for an old IPsec entry when it is disabled or removed #2719
- Manage active SPDs on CARP secondary during sync #2303
- Add an option to force IPsec to reload on failover, which is needed in some cases for IPsec to fail from one interface to another. #2896

OpenVPN

- OpenVPN can accept attributes from RADIUS via avpairs for things like inacl, outacl, dns-server, routes
- OpenVPN checkbox for "topology subnet" to use one IP per client in tun mode
- OpenVPN local/remote network boxes can accept multiple comma-separated networks
- OpenVPN status for SSL/TLS server instances can now display the routing table for the VPN instance
- OpenVPN now allows selecting "localhost" as the interface
- Gateways are created for assigned OpenVPN server instances as well as clients
- OpenVPN instances can run on the same port on different interfaces

- OpenVPN status page now has service controls to show the status of the daemon running each instance, and allow for stop/start/restart from that page
- Changed wording of the error displayed when a daemon is not running or the management interface of OpenVPN cannot be reached for an instance
- OpenVPN client-specific Override cleanup fixes
- Fixed double-click to edit of OpenVPN Client-Specific Overrides

NAT/Firewall Rules/Alias

- Aliases separated into tabs for Hosts, Ports, and URLs to improve manageability
- NAT reflection options re-worded to be less confusing
- Adjustable source tracking timeout for Sticky connections
- Firewall rules now support matching on ECE and CWR TCP flags
- Filtering on ECE and CWR TCP flags is now possible
- Added ICMP to protocol list when creating rdr (port forward) rules
- Keep proper positioning of duplicated outbound NAT rules #1118
- When using the + at the top of Outbound NAT rules, add the rule to the top of the list and not the bottom
- Fix ordering of interface group rules in the ruleset #2837
- Track time and user@host which created or updated a firewall, NAT port forward, or outbound NAT rule. If timestamp records are present, display them at the bottom of the rule page when editing. Have the created time/user pre-filled for automated rules such as NAT port forward associated rules and the switch from automatic to manual outbound NAT
- Fix generation of manual outbound NAT rules so that localhost and VPN rules are not unnecessarily duplicated
- Prevent using “block” for an alias name, as it is a pf reserved keyword
- Allow TCP flags to be used on block or reject rules, since they are also valid there
- Updates/fixes to DSCP handling
- Allow advanced options state-related parameters to be used for TCP, UDP and ICMP – Formerly only allowed on TCP
- Respect ports found in rules when policy route negation rules are made, #3173
- Do not include disabled OpenVPN networks in generated policy route negation rules

Certificates

- Improved denoting of certificate purposes in the certificate list
- Imported CRLs can be edited and replaced
- Can set digest algorithm for CA/Certs (sha1, sha256, etc)
- Default digest algorithm is now SHA256
- Show CA and certificate start and end dates in the their listings
- Correct tooltip description when adding a certificate #3017

- Relax input validation on a CA/Cert description since it is only used cosmetically in pfSense and not in the actual CA/cert subject
- Allow removing blank/empty CA and Cert entries

Logging

- More system log separation, Gateways, Routing, Resolver split into their own tabs
- Firewall logs can now be filtered by many different criteria
- Firewall logs can be sorted by any column
- Firewall logs can optionally show the matching rule description in a separate column or in between rows
- Firewall logs now show an indicator icon if the direction of a log entry is OUT rather than IN
- Add popup DNS resolution method to firewall log view
- Reduced logging output from IGMP proxy
- Reduced logging output from DynDNS
- Relocated filterdns logs to the resolver log file/tab
- Relocated DHCP client logs to the DHCP tab
- Fix system script logging so the correct script filename is printed in the log, rather than omitting the script name entirely
- Add independent logging choices to disable logging of bogon network rules and private network rules. Add upgrade code to obey the existing behavior for users (if default block logging was disabled, so is bogon/private rule blocking)
- Add a checkbox to disable the lighttpd log for people who don't want their system log full of messages from lighttpd in some cases where they are filling the log unnecessarily

Notifications

- Add the ability to disable Growl or SMTP notifications but keep their settings intact, so the mail settings can be used for other purposes (packages, etc)
- Add a test button to selectively test Growl or SMTP notifications without re-saving settings
- Do not automatically generate a test notification on saving notification settings, as there are now individual test buttons

High Availability (CARP, pfSync, XML-RPC)

- High Availability Synchronization options (Formerly known as "CARP Settings" under Virtual IPs Promoted to its own menu entry, System > High Avail. Sync
  - This is to make it easier to find, as well as make its purpose more clear. "CARP" is a part of High Availability, as is XMLRPC/pfsync state synchronization, but it's a bit of a misnomer to refer to the sync settings as CARP
- Ensure that the user does not remove only the last IP alias needed for a CARP VIP in an additional subnet
- Disable pfsync interface when state synchronization is not in use
- Fixed issues with DHCP server config synchronization ordering on secondary nodes #2600

- Restart OpenVPN servers when CARP transitions to master (clients were already restarted), otherwise if CARP was disabled, the servers would never recover
- Removed the automatic pfsync rule, since the documentation always recommends adding it manually, and to add it behind the scenes with no way to block it can be counter-productive (and potentially insecure). **If the documentation was not followed and a pfsync or allow all rule was not added on the sync interface, then state synchronization may break after this upgrade. Add an appropriate rule to the sync interface and it will work again.**
- Allow XMLRPC to sync IP Alias VIPs set to Localhost for their interface
- In DHCP leases view, use the internal interface name (lan/opt1/etc) for the failover pool name, rather than a number. In certain cases the number can get out of sync between the two nodes, but the interface names will always match
- Print the user-configured interface description next to the DHCP failover pool name, rather than only the internal name (lan/opt1/etc)
- Add option to synchronize authentication servers (RADIUS, LDAP) via XMLRPC

NanoBSD

- Fixes for conf_mount_ro/conf_mount_rw reference checking/locking
- Diag > NanoBSD now has button to switch media between read/write and read-only
- Diag > NanoBSD now has a checkbox option to keep the media read/write
- Fixed an issue with NanoBSD time zones not being properly respected by all processes the first reboot after a firmware upgrade

DHCP Server

- DHCP can support multiple pools inside a single subnet, with distinct options per pool
- DHCP can allow/deny access to a DHCP pool by partial (or full) MAC address
- DHCP static mappings can have custom settings for gateway, DNS, etc
- DHCP static mappings can optionally have a static ARP entry created
- Fix Dynamic DNS updates from DHCP (ISC changed the config layout and requires zone declarations)
- When crafting DHCP Dynamic DNS zones, do not use invalid DNS servers for the IP type (e.g. skip IPv6 DNS servers, because the DHCP daemon rejects them)
- Added a config backup section choice for DHCPv6

Traffic Shaper

- Schedules can now be used with limiters
- Traffic shaper queues view updated
- CoDel AQM Shaper Discipline
- Allow PRIQ queues to be deleted. #3037
- Limiters now allow the user to set the mask they want to use, rather than assuming masking will always be per-IP. This allows per-subnet limits and similar

- Limiters now allow setting masking for IPv6
- Limiters now allow setting a burst size. This will pass X amount of data (TOTAL, NOT a rate) after an idle period before enforcing the limit

DNS Forwarder

- In DNS forwarder, DNS query forwarding section with options for sequential and require domain
- Allow a null forwarding server in DNS Forwarder domain overrides to ensure that queries stay local and never go outside the firewall
- Add DNS Forwarder option to not forward private reverse lookups
- DNS Forwarder domain overrides can now specify a source address for the query, to help resolve hostnames over VPN tunnels
- DNS Forwarder now can change the port upon which it listens, for better cohabitation with other DNS software such as tinydns or unbound, if both are needed
- DNS Forwarder now has an option to select the interfaces/IP Addresses upon which it will respond to queries
- DNS Forwarder can now be set to only bind to specific IPv4 IPs (the underlying software, dnsmasq, does not support selectively binding to IPv6 IPs)
- Improved handling of some dnsmasq custom config options

User Manager

- Configurable RADIUS authentication timeout in User Manager
- Print the error message from LDAP in the log for a bind failure. Helps track down reasons for authentication failures
- Re-enable admin user if it's disabled when 'Reset webConfigurator password' option is used. Fixes #2877
- Restrict maximum group name length to 16 characters or less (OS restriction)
- Added option to UTF-8 encode LDAP parameters to improve handling of international characters
- CDATA protected LDAP fields in config to avoid invalid XML with international characters

DynDNS

- Fixed handling of DynDNS 25-day update and add ability to configure update interval
- Added DynDNS No-IP Free Account Support
- Add AAAA support to RFC2136 updates
- Add cached IP support to RFC2136, add GUI button to force update for single host
- Fix double click row to edit for RFC2136
- Add option to RFC2136 to find/use the public IP if the interface IP is private. (Off by default to preserve existing behavior on upgrade)
- Add server IP column and cached IP display to RFC2136 host list
- Include RFC2136 hosts in DNS rebinding checks
- Include both dyndns and RFC2136 hosts in referer check

Graphs

- Add ability to reverse-resolve IPs on Status > Traffic Graph in the rate table
- Add ability to filter local or remote IPs on Status > Traffic Graph in the rate table
- Change maximum values for RRD throughput to account for 10G links. Previous maximums would have caused blank spots on the graph during periods of high throughput
- Fixes to RRD data resolution/retention
- Added RRD Graph for mbuf clusters
- Changed default RRD graph colors to be more visually distinct to help avoid ambiguity between multiple values on the same graph

Misc

- Add option to the packet capture page to control whether or not promiscuous mode is used on the NIC. Certain drivers have issues with promiscuous mode
- Make parent interface and all VLANs share MTU #2786
- Fix cellular signal strength indicator
- Fix PPP config cleanup when removing an interface #2758
- Disallow adding IP Alias or CARP VIP that would be the network or broadcast address of a subnet
- Diagnostics > Sockets page to show open network sockets on the firewall
- Diagnostics > Test Port page to perform a simple TCP connection test to see if a port is open
- The pftop page has additional options to display more detailed information and sort it
- Fixed conflict between static IP and static route in the same subnet #2039
- Do not apply static ARP entries to disabled interfaces #1988
- Do not allow bridge members to be assigned to itself #1153
- Changed Diag > Ping to use more available source addresses (CARP VIPs, IP Alias VIPs, OpenVPN interfaces, IPv6 Link-Local IPs)
- Changed Diag > Traceroute to use more available source addresses (CARP VIPs, IP Alias VIPs, OpenVPN interfaces, IPv6 Link-Local IPs)
- Changed shell prompt to not force background color, to be kinder to those not using black as a background in their terminal
- Add a field to allow rejecting DHCP leases from a specific upstream DHCP server. #2704
- Updated the help system to handle some recent added files for 2.x and clean out some old/obsolete files
- Allow selecting “Localhost” as an interface for IP Alias VIPs - this way IP Alias VIPs may be used for binding firewall services (e.g. Proxy, VPN, etc) in routed subnets without burning IPs for CARP unnecessarily
- Updated list of mobile service providers
- Fix max length for wpa passphrase. A 64-char passphrase would be rejected by hostapd and leave an AP in an open state #3034
- Added MSS clamping to the setup wizard
- Add a setting to configure the filterdns hostname resolution interval (defaults to 300s, 5 minutes)

- Omit IP mismatch warnings (e.g. behind a port forward, VPN IP, etc) if HTTP_REFERER protection is disabled
- Fixes for selecting/detecting PPP devices such as 3G/4G modems
- Rather than doing auto-detection to find serial PPP devices, use a glob when listing potential PPP serial devices
- Prevent sshlockout from a crash/coredump if a format string like %s is present in the buffer
- Fix SMART to see adaX devices
- Fix SMART interpretation of output from SCSI devices
- Fixed display of user SSH keys when present
- Updated p0f database from FreeBSD
- Fix UPnP Interface name selection to show the configured description entered by the user
- Allow setting the external UPnP interface (must be default route WAN)
- Fix Diag > Tables AJAX fadeOut after deletion for rows with CIDR mask format
- Improve Diagnostics > Routes to fetch output via AJAX and have configurable filtering and sizes. Improves handling of large routing tables, such as a full BGP feed
- When deleting or renaming a virtual server from the Load Balancer (relayd) manually clean up the NAT rules it leaves behind to avoid conflicts
- Many, many bug fixes
- Various fixes for typos, formatting, input validation, etc

SH/PHP Shell Scripts

- Git package for gitsync is now pulled in as a pfSense-style PBI package
- Shell scripts added to enable/disable CARP:

  pfSsh.php playback enablecarp pfSsh.php playback disablecarp

- Shell scripts to add and remove packages from the command line:

```
pfSsh.php playback installpkg "Some Package"
pfSsh.php playback uninstallpkg "Some Package"
pfSsh.php playback listpkg
```

- Added shell script to remove shaper settings:

```
pfSsh.php playback removeshaper
```

- Add shell script to control services from the command line:

```
pfSsh.php playback svc start <service name>
pfSsh.php playback svc restart <service name>
pfSsh.php playback svc stop <service name>
```

- Add a simple CLI mail script capable of sending an SMTP message using echo/piped input (uses SMTP notification settings for server details):

```
ifconfig -a | mail.php -s"ifconfig output"
```

- Added a script to convert a user's filesystem from device names to UFS labels, for easier portability in case the disk device changes names (e.g. adX to adY, adX to daY, or adX to adaX). ONLY FOR FULL INSTALLS. NanoBSD already uses labels.

```
/usr/local/sbin/ufslabels.sh
```

3.3.36 2.0.3 New Features and Changes

pfSense® software 2.0.3 is a maintenance release with some bug fixes since 2.0.2 release. It is possible to upgrade from any previous release to 2.0.3.

Because this release shortly followed after 2.0.2, review the *2.0.2 New Features and Changes* document as well.

The changelog for pfSense 2.0.3-RELEASE follows.

Security Advisories

- Updated to OpenSSL 0.9.8y to address FreeBSD-SA-13:03

PPP

- Fix obtaining DNS servers from PPP type WANs (PPP, PPPoE, PPTP, L2TP)

Captive Portal

- Fix Captive Portal Redirect URL trimming
- Voucher sync fixes
- Captive portal pruning/locking fixes
- Fix problem with fastcgi crashing

OpenVPN

- Clear the route for an OpenVPN endpoint IP when restarting the VPN, to avoid a situation where a learned route from OSPF or elsewhere could prevent an instance from restarting properly
- Always clear the OpenVPN route when using shared key, no matter how the tunnel network "CIDR" is set
- Use the actual OpenVPN restart routine when starting/stopping from services rather than killing/restarting manually
- Allow editing an imported CRL, and refresh OpenVPN CRLs when saving. #2652
- Fix interface assignment descriptions when using > 10 OpenVPN instances

Logging

- Put syslogd into secure mode so it refuses remote syslog messages
- If syslog messages are in the log, and the hostname does not match the firewall, display the supplied hostname
- Fix PPP log display to use the correct log handling method
- Run IPsec logs through htmlspecialchars before display to avoid a potential persistent XSS from racoon log output (e.g. username)

Traffic Shaper

- Fix editing of traffic shaper default queues. #1995
- Fix wording for VoIP address option in the shaper. Add rule going the other direction to catch connections initiated both ways

Dashboard & General GUI

- Use some tweaks to PHP session management to prevent the GUI from blocking additional requests while others are active
- Remove cmd_chain.inc and preload.php to fix some issues with lighttpd, fastcgi, and resource usage
- Firmware settings manifest (Site list) now bolds and denotes entries that match the current architecture, to help avoid accidental cross-architecture upgrades
- Add header to DHCP static mappings table
- When performing a factory reset in the GUI, change output style to follow halt.php and reboot.php so the shutdown output appears in the correct location on the page
- Better validation of parameters passed during S.M.A.R.T. operations for testing HDDs
- Fixed SNMP interface binding glitch (Setting was active but not reflected when viewed in GUI)
- Add a new class called addgatewaybox to make it easier to respect custom themes #2900

Console Menu Changes

- Correct accidental interface assignment changes when changing settings on the console menu
- Console menu option 11 now kills all active PHP processes, kills lighttpd, and then restarts the GUI. This is a more effective way to restart the GUI since if a PHP process is hung, restarting lighttpd alone will not recover from that
- Fix port display after LAN IP reset

Misc Changes

- Change how the listening address is passed to miniupnpd, the old method was resulting in errors for some users
- Fix "out" packet count reporting
- Be a little smarter about the default kernel in rare cases where we cannot determine what was in use
- Pass -S to tcpdump to avoid an increase in memory consumption over time in certain cases
- Minimise rewriting of /etc/gettytab (https://forum.netgate.com/post/51581)
- Make is_pid_running function return more conistent results by using isvalidpid
- Fix ataidle error on systems that have no ATA HDD. #2739
- Update Time Zone database zoneinfo to 2012.j to pick up on recent zone/DST/etc changes
- Fix handling of LDAP certificates, the library no longer properly handles files with spaces in the CA certificate filename
- Bring in the RCFILEPREFIX as constant fixes from HEAD, since otherwise rc.stop_packages was globbing in the wrong dir and executing the wrong scripts. Also seems to have fixed the "bad fd" error
- NTP restart fixes
- Gitsync now pulls in git package from pfSense package repository rather than FreeBSD
- Fixed handling of RRD data in config.xml backups when exporting an encrypted config #2836
- Moved apinger status to /var/run instead of /tmp
- Fixes for FTP proxy on non-default gateway WANs
- Fixes for OVA images
- Use new pfSense repository location (http://github.com/pfsense/pfsense/)
- Add patch to compensate apinger calculation for down gateways by time taken from other tasks like rrd/status file/etc

lighttpd changes

- Improve tuning of lighttpd and php processes
- Use separate paths for GUI and Captive Portal fastcgi sockets
- Always make sure php has its own process manager to make lighttpd happy
- Make mod_fastcgi last to have url.rewrite work properly
- Enable mod_evasive if needed for Captive Portal
- Simplify lighttpd config
- Send all lighttpd logs to syslog

Binary changes

- dnsmasq to 2.65
- rsync to 3.0.9
- links 2.7
- rrdtool to 1.2.30
- PHP to 5.2.17_13
- OpenVPN 2.2 stock again (Removed IPv6 patches since those are only needed on 2.1 now)
- Fix missing "beep" binary on amd64
- Fix potential issue with IPsec routing of client traffic
- Remove lighttpd spawnfcgi dependency
- Add splash device to wrap_vga kernels (It's in GENERIC so full installs already have it). #2723

filterdns

- Correct an issue with unallocated structure
- Avoid issues with pidfiles being overwritten, lock the file during modifications
- Make filterdns restartable and properly cleanup its tables upon exit or during a reconfiguration

dhcpleases

- Correct use after free and also support hostnames with other DNS suffix
- Reinit on any error rather than just forgetting. Also the difftime checks are done after having complete view, no need to do them every time
- Typo fixes
- Log that a HUP signal is being sent to the pid file submitted by argument
- Prevent bad parsing of empty hostnames in lease file. Add an f option to run dhcplease in foreground. The only option needed while in foreground is h parameter and the only usable one as well

3.3.37 2.0.2 New Features and Changes

pfSense® software 2.0.2 is a maintenance release with some bug and security fixes since 2.0.1 release. It is possible to upgrade from any previous release to 2.0.2.

What follows is a mostly-complete changelog for pfSense 2.0.2-RELEASE

FreeBSD Security Advisories

Base OS updated to 8.1-RELEASE-p13 to address the following FreeBSD Security Advisories:

- FreeBSD-SA-12:01.openssl (v1.0/v1.1) http://security.freebsd.org/advisories/FreeBSD-SA-12:01.openssl.asc
- FreeBSD-SA-12:02.crypt http://security.FreeBSD.org/advisories/FreeBSD-SA-12:02.crypt.asc
- FreeBSD-SA-12:04.sysret (v1.0/v1.1) http://security.FreeBSD.org/advisories/FreeBSD-SA-12:04.sysret.asc
- FreeBSD-SA-12:07.hostapd https://www.freebsd.org/security/advisories/FreeBSD-SA-12:07.hostapd.asc
- NOTE: FreeBSD-SA-12:03.bind, FreeBSD-SA-12:05.bind, and FreeBSD-SA-12:06.bind do not apply to us, since we do not use nor include bind. FreeBSD-SA-12:08.linux does not apply since we do not use nor include the Linux compatibility layer of FreeBSD.

PPTP

- Added a warning to PPTP VPN configuration page: **PPTP is no longer considered a secure VPN technology because it relies upon MS-CHAPv2 which has been compromised. Be aware that intercepted traffic can be decrypted by a third party, so it should be considered unencrypted. We advise migrating to another VPN type such as OpenVPN or IPsec.**
  - More information on this can be found at https://isc.sans.edu/diary/End+of+Days+for+MS-CHAPv2/13807 and https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2/
- Fix reference to PPTP secondary RADIUS server shared secret.
- PPTP upgrade fixes.

NTP Changes

- OpenNTPD was dropped in favor of the ntp.org NTP daemon, used by FreeBSD.
- Status page added (Status > NTP) to show status of clock sync
- NTP logging fixed.
- NOTE: ntpd will bind/listen to all interfaces by default, and it has to in order to receive replies. Selective interface binding may still be used to control which IP addresses will accept traffic, but be aware that the default behavior has changed.

Dashboard & General GUI Fixes

- Various fixes for typos, wording, and so on.
- Do not redirect on saving services status widget.
- Don't use $pconfig in widgets, it has unintended side effects.
- Fix display of widgets with configuration controls in IE.
- Changed some padding/margin in the CSS in order to avoid wrapping the menu.
- #2165 Change to embed to prevent IE9 from misbehaving when loading the Traffic Graph page

OpenVPN Fixes

- Safer for 1.2.3 upgrades to assume OpenVPN interface == any, since 1.2.3 didn't have a way to bind to an interface. Otherwise people accepting connections on opt interfaces on 1.2.3 will break on upgrade until the proper interface is selected in the GUI
- Don't ignore when multiple OpenVPN DNS, NTP, WINS, etc servers were specified in 1.2.3 when upgrading. 1.2.3 separated by ;, 2.x uses separate vars.
- Fix upgrade code for 1.2.3 with assigned OpenVPN interface.
- Fix LZO setting for Upgraded OpenVPN (was turning compression on even if old config had it disabled.)
- Be more intelligent when managing OpenVPN client connections bound to CARP VIPs. If the interface is in BACKUP status, do not start the client. Add a section to rc.carpmaster and rc.carpbackup to trigger this start/stop. If an OpenVPN client is active on both the master and backup system, they will cause conflicting connections to the server. Servers do not care as they only accept, not initiate.

IPsec fixes

- Only do foreach on IPsec p2's if it's actually an array.
- #2201 Don't let an empty subnet into racoon.conf, it can cause parse errors.
- #2201 Reject an interface without a subnet as a network source in the IPsec Phase 2 GUI.
- Add routes even when IPsec is on WAN, as WAN may not be the default gateway.
- #1986 Revamped IPsec status display and widget to properly account for mobile clients.
- Fixed a bug that caused the IPsec status and widget to display slowly when mobile clients were enabled.

User Manager Fixes

- #2066 Improve adding/removing of users accounts to the underlying OS, especially accounts with a numeric username.
- Include admin user in bootup account sync
- Fix permission and certificate display for the admin user
- Fix ssh key note to refer to DSA not just RSA since both work.
- ":" chars are invalid in a comment field, filter them out.
- When renaming a user, make sure to remove the previous user or it gets left in /etc/passwd.
- #2326 Do not allow empty passwords since this might cause problems for some authentication servers like LDAP.

Captive Portal Fixes

- Take routing table into account when figuring out which IP address to use for talking to CP clients.
- Prevent browser auto-fill username and password on voucher config, as it can interfere with the settings being properly saved if sync isn't fully configured, which this can make happen accidentally.
- Correct the Called-Station-Id attribute setting to be the same on STOP/START packets
- Correct the Called-Station-Id attribute setting to be consistent on the data sent
- #2082 Correct the log to display the correct information about an existing session
- #2052 Remove duplicate rule
- Fix which roll to write when writing the active voucher db
- Always load ipfw when enabling CP to ensure the pfil hooks are setup right
- #2378 Fix selection of CP interfaces when using more than 10 opt interfaces.
- Strengthen voucher randomization.

NAT/Firewall Rules/Alias Fixes

- #2327 Respect the value of the per-rule "disable reply-to" checkbox.
- #1882 Fix an invalid pf rule generated from a port forward with dest=any on an interface with ip=none
- #2163 1:1 Reflection fixes for static route subnets and multiple subnets on the same interface.
- Better validation on URL table alias input from downloaded files.
- #2293 Don't put an extra space after "pass" when assuming it as the default action or later tests will fail to match this as a pass rule.
- Update help text for Host aliases to indicate FQDNs are allowed.
- #2210 Go back to scrub rather than "scrub in", the latter breaks MSS clamping for egress traffic the way we use it.
- Fix preservation of the selection of interfaces on input errors for floating rules.
- Fix URL table update frequency box.
- Fix input validation for port forwards, Local Port must be specified.
- Added a setting to increase the maximum number of pf tables, and increased the default to 3000.
- Properly determine active GUI and redirect ports for anti-lockout rule, for display and in the actual rule.
- Handle loading pf limits (timers, states, table/entry limits, etc) in a separate file to avoid a chicken-and-egg scenario where the limits would never be increased properly.

Interface/Bridging Fixes

- Correct checking if a gif is part of bridge so that it actually works correctly adding a gif after having created it on bootup
- Use the latest functions from pfSense module for getting interface list
- Use the latest functions from pfSense module for creating bridges
- Implement is_jumbo_capable in a more performant way. This should help with large number of interfaces
- Since the CARP interface name changed to "vipN" from "carpN", devd needs to follow that change as well.
- #2242 Show lagg protocol and member interfaces on Status > Interfaces.
- #2212 Correctly stop dhclient process when an interface is changed away from DHCP.
- Fixed 3G SIM PIN usage for Huawei devices
- Properly obey MTU set on Interface page for PPP type WANs.

Other Misc. Fixes

- #2057 Add a checkbox that disables automatically generating negate rules for directly connected networks and VPNs.
- Mark "Destination server" as a required field for DHCP Relay
- Clarify the potential pitfalls when setting the Frequency Probe and Down parameters.
- Add a PHP Shell shortcut to disable referer check (playback disablerefercheck)
- #2040 Make Wireless Status tables sortable
- #2068 Fix multiple keys in a file for RFC2136 dyndns updates.
- Check to see if the pid file exists before trying to kill a process
- #2144 Be smarter about how to split a Namecheap hostname into host/domain.
- Add a small script to disable APM on ATA drives if they claim to support it. Leaving this on will kill drives long-term, especially laptop drives, by generating excessive Load Cycles. The APM bit set will persist until the drive is power cycled, so it's necessary to run on each boot to be sure.
- #2158 Change SNMP binding option to work on any eligible interface/VIP. If the old bindlan option is there, assume the lan interface for binding.
- Fix reference to PPTP secondary RADIUS server shared secret.
- PPTP upgrade fixes.
- #2147 Add button to download a .p12 of a cert+key.
- #2233 Carry over the key length on input errors when creating a certificate signing request.
- #2207 Use PHP's built-in RFC 2822 date format, rather than trying to make our own.
- Allow specifying the branch name after the repository URL for gitsync command-line arguments and remove an unnecessary use of the backtick operator.
- Correct send_multiple_events to conform with new check_reload_status behaviour
- Do not wipe logs on reboot on full install
- Set FCGI_CHILDREN to 0 since it does not make sense for php to manage itself when lighttpd is doing so. This makes it possible to recover from 550-Internal… error.

- Support for xmlrpcauthuser and xmlrpcauthpass in $g.
- Fix Layer 7 pattern upload, button text check was incorrect.
- Correct building of traffic shaping queue to not depend on parent mask
- #2239 Add alias support to static routes
- Use !empty instead of isset to prevent accidental deletion of the last used repository URL when firmware update gitsync settings have been saved without a repository URL.
- Better error handling for crypt_data and also better password argument handling
- Stop service needs to wait for the process to be stopped before trying to restart it.
- Use a better default update url
- Fix missing description in rowhelper for packages.
- #2402, #1564 Move the stop_packages code to a function, and call the function from the shell script, and call the function directly for a reboot.
- #1917 Fix DHCP domain search list
- Update Time Zone zoneinfo database using latest zones from FreeBSD
- Handle HTTPOnly and Secure flags on cookies
- Fixed notifications for firmware upgrade progress
- Removed an invalid declaration that considered 99.0.0.0/8 a private address.
- Fixed redirect request for IE8/9
- #1049 Fix crashes on NanoBSD during package removal/reinstall. Could result in the GUI being inaccessible after a firmware update.
- Fix some issues with upgrading NanoBSD+VGA and NanoBSD+VGA Image Generation
- Fix issues upgrading from systems with the old “Uniprocessor” kernel which no longer exists.
- Fix a few potential XSS/CSRF vectors.
- Fixed issue with login page not showing the correct selected theme in certain configurations.
- Fix limiters+multi-wan

Binary/Supporting Program Updates

- Some cleanup to reduce overall image size
- Fixes to ipfw-classifyd file reading and handling
- Updated miniupnpd
- ISC DHCPD 4.2.4-P1
- mdp5 upgraded to 5.6
- pftop updated
- lighttpd updated to 1.4.32, for CVE-2011-4362 and CVE-2012-5533.

3.3.38 2.0.1 New Features and Changes

This is a maintenance release with some bug and security fixes since 2.0 release. It is possible to upgrade from any previous release to 2.0.1.

For those who use the built in certificate manager, pay close attention to the notes below on a potential security issue with those certificates.

Change Log

The following changes were made after 2.0-RELEASE and were included in 2.0.1-RELEASE.

- Improved accuracy of automated state killing in various cases (#1421)
- Various fixes and improvements to relayd
  - Added to Status > Services and widget
  - Added ability to kill relayd when restarting (#1913)
  - Added DNS load balancing
  - Moved relayd logs to their own tab
  - Fixed default SMTP monitor syntax and other send/expect syntax
- Fixed path to FreeBSD packages repo for 8.1
- Various fixes to syslog:
  - Fixed syslogd killing/restarting to improve handling on some systems that were seeing GUI hangs resetting logs
  - Added more options for remote syslog server areas
  - Fixed handling of 'everything' checkbox
  - Moved wireless to its own log file and tab
- Removed/silenced some irrelevant log entries
- Fixed various typos
- Fixes for RRD upgrade/migration and backup (#1758)
- Prevent users from applying NAT to CARP which would break CARP in various ways (#1954)
- Fixed policy route negation for VPN networks (#1950)
- Fixed "Bypass firewall rules for traffic on the same interface" (#1950)
- Fixed VoIP rules produced by the traffic shaper wizard (#1948)
- Fixed uname display in System Info widget (#1960)
- Fixed LDAP custom port handling
- Fixed Status > Gateways to show RTT and loss like the widget
- Improved certificate handling in OpenVPN to restrict certificate chaining to a specified depth – CVE-2011-4197
- Improved certificate generation to specify/enforce type of certificate (CA, Server, Client) – CVE-2011-4197
- Clarified text of serial field when importing a CA (#2031)
- Fixed MTU setting on upgrade from 1.2.3, now upgrades properly as MSS adjustment (#1886)
- Fixed Captive Portal MAC passthrough rules (#1976)

- Added tab under Diagnostics > States to view/clear the source tracking table if sticky is enabled
- Fixed CARP status widget to properly show “disabled” status.
- Fixed end time of custom timespan RRD graphs (#1990)
- Fixed situation where certain NICs would constantly cycle link with MAC spoofing and DHCP (#1572)
- Fixed OpenVPN ordering of client/server IPs in Client-Specific Override entries (#2004)
- Fixed handling of OpenVPN client bandwidth limit option
- Fixed handling of LDAP certificates (#2018, #1052, #1927)
- Enforce validity of RRD graph style
- Fixed crash/panic handling so it will do textdumps and reboot for all, and not drop to a db> prompt.
- Fixed handling of hostnames in DHCP that start with a number (#2020)
- Fixed saving of multiple dynamic gateways (#1993)
- Fixed handling of routing with unmonitored gateways
- Fixed Firewall > Shaper, By Queues view
- Fixed handling of spd.conf with no phase 2’s defined
- Fixed synchronization of various sections that were leaving the last item on the slave (IPsec phase 1, Aliases, VIPs, etc)
- Fixed use of quick on internal DHCP rules so DHCP traffic is allowed properly (#2041)
- Updated ISC DHCP server to 4.2.3 (#1888) – this fixes a denial of service vulnerability in dhcpd.
- Added patch to mpd to allow multiple PPPoE connections with the same remote gateway
- Lowered size of CF images again fix newer and ever-shrinking CF cards.
- Clarified text for media selection (#1910)

Notes for certificate generation vulnerability

Certificates generated with the built-in certificate manager in all 2.0 versions prior to 2.0.1 are excessively permissive for non-CA certificates. These certificates can be used as a certificate authority, meaning a user can use their own certificate to create chained certificates. We have defaulted OpenVPN on 2.0.1 and newer versions to not accept chained certificates, which mitigates this. However, if untrusted users have certificates generated from 2.0 release, we suggest re-generating all certificates and issuing new ones. Certificates generated by easy-rsa and imported into 2.0 are not affected. If using certificates generated on pfSense® for other purposes, revoke those and issue new certificates generated on 2.0.1. A CRL must be utilized in that case. To be on the safe side, start from scratch with a new CA and certificates after deleting all existing ones. Thanks to Florent Daigniere for bringing this issue to our attention and helping confirm our resolution.

Upgrade considerations

It is very important to read the *Upgrade Guide* before performing an upgrade for those still on 1.2.x versions.

3.3.39 2.0 New Features and Changes

This is a partial list of the new features and major changes in the pfSense® software 2.0 release.

Operating System

- Based on FreeBSD 8.1 release.
- i386 and amd64 variants for all install types (full install, nanobsd/embedded, etc.)
- USB memstick installer images available

Interfaces

- GRE tunnels
- GIF tunnels
- 3G support
- Dial up modem support
- Multi-Link PPP (MLPPP) for bonding PPP connections (ISP/upstream must also support MLPPP)
- *LAGG Interfaces*
- Interface groups
- IP Alias type Virtual IPs
- IP Alias VIPs can be stacked on CARP VIPs to go beyond the 255 VHID limit in deployments that need very large numbers of CARP VIPs.
- QinQ VLANs
- Can use Block Private Networks / Block Bogon Networks on any interface
- All interfaces are optional except WAN
- All interfaces can be renamed, even LAN/WAN
- Bridging enhancements - can now control all options of if_bridge, and assign bridge interfaces

Gateways/Multi-WAN

- Gateways, including dynamic gateways, are specified under System > Routing
- Gateways can have custom monitor IPs
- Gateways can have a custom weight, allowing load balancing to have ratios between WANs of different speeds
- Gateways can have custom latency, loss, and downtime trigger levels.
- Gateway monitoring via icmp is now configurable.
- Multiple gateways may exist per interface

- Multi-WAN is now handled via gateway groups
- Gateway groups can include multiple tiers with any number of gateways on each, for complex failover and load balancing scenarios.

General Web GUI

- Set to HTTPS by default, HTTP redirects to HTTPS port
- Dashboard and widgets added
- System > Advanced screen split into multiple tabs, more options available.
- SMTP email alerts and growl alerts
- New default theme - pfsense_ng
- Some community-contributed themes added
- Contextual help available on every page in the web interface, linking to a webpage containing help and documentation specific to that page.
- Help menu for quick access to online resources (forum, docs, paid support, etc.)

Aliases

- Aliases may be nested (aliases in aliases)
- Alias autocomplete is no longer case sensitive
- IP Ranges in Aliases
- More Alias entries supported
- Bulk Alias importing
- URL Aliases
- URL Table Aliases - uses a pf persist table for large (40,000+) entry lists

Firewall

- Traffic shaper rewritten - now handles any combination of multi-WAN and multi-LAN interfaces. New wizards added.
- Layer7 protocol filtering
- *EasyRule - add firewall rules from log view (and from console!)*
- Floating rules allow adding non-interface specific rules
- Dynamically sized state table based on amount of RAM in the system
- More Advanced firewall rule options
- FTP helper now in kernel
- TFTP proxy
- Schedule rules are handled in pf, so they can use all the rule options.
- State summary view, report shows states grouped by originating IP, destination IP, etc.

NAT

- All of the NAT screens were updated with additional functionality
- Port forwards can now handle create/update associated firewall rules automatically, instead of just creating unrelated entries.
- Port forwards can optionally use “rdr pass” so no firewall rule is needed.
- Port forwards can be disabled
- Port forwards can be negated (“no rdr”)
- Port forwards can have source and destination filters
- NAT reflection improvements, including NAT reflection for 1:1 NAT
- Per-entry NAT reflection overrides
- 1:1 NAT rules can specify a source and destination address
- 1:1 NAT page redesigned
- Outbound NAT can now translate to an address pool (Subnet of IPs or an alias of IPs) of multiple external addresses
- Outbound NAT rules can be specified by protocol
- Outbound NAT rules can use aliases
- Improved generation of outbound NAT rules when switching from automatic to manual.

IPsec

- Multiple IPsec p2’s per p1 (multiple subnets)
- IPsec xauth support
- IPsec transport mode added
- IPsec NAT-T
- Option to push settings such as IP, DNS, etc, to mobile IPsec clients (mod_cfg)
- Mobile IPsec works with iOS and Android (Certain versions, see *IPsec Remote Access VPN Example Using IKEv1 with Xauth*)
- More Phase 1/2 options can be configured, including the cipher type/strength
- ipsec-tools version 0.8

User Manager

- New user manager, centralizing the various user configuration screens previously available.
- Per-page user access permissions for administrative users
- Three built-in authentication types - local users, LDAP and RADIUS.
- Authentication diagnostics page

Certificate Manager

- Certificate manager added, for handling of IPsec, web interface, user, and OpenVPN certificates.
- Handles creation/import of Certificate Authorities, Certificates, Certificate Revocation lists.
- Eliminates the need for using command line tools such as EasyRSA for managing certificates.

OpenVPN

- OpenVPN wizard guides through making a CA/Cert and OpenVPN server, sets up firewall rules, and so on. Greatly simplifies the process of creating a remote access OpenVPN server.
- OpenVPN filtering - an OpenVPN rules tab is available, so OpenVPN interfaces don't have to be assigned to perform filtering.
- OpenVPN client export package - provides a bundled Windows installer with certificates, Viscosity export, and export of a zip file containing the user's certificate and configuration files.
- OpenVPN status page with connected client list – can also kill client connections
- User authentication and certificate management
- RADIUS and LDAP authentication support

Captive Portal

- Voucher support added
- Multi-interface capable
- Pass-through MAC bandwidth restrictions
- Custom logout page contents can be uploaded
- Allowed IP addresses bandwidth restrictions
- Allowed IP addresses supports IP subnets
- "Both" direction added to Allowed IP addresses
- Pass-through MAC Auto Entry - upon successful authentication, a pass-through MAC entry can be automatically added.
- Ability to configure calling station RADIUS attributes

Wireless

- Virtual AP (VAP) support added
- more wireless cards supported with the FreeBSD 8.1 base

Server Load Balancing

- relayd and its more advanced capabilities replace slbd.

Other

- L2TP VPN added
- DNS lookup page added
- PFTop and Top in GUI - realtime updates
- Config History now includes a diff feature
- Config History has download buttons for prior versions
- Config History has mouseover descriptions
- CLI filter log parser (/usr/local/bin/filterparser)
- Switched to PHP 5.2.x
- IGMP proxy added
- Multiple Dynamic DNS account support, including full multi-WAN support and multi-accounts on each interface.
  - DynDNS Account Types supported are:
    - DNS-O-Matic
    - DynDNS (dynamic)
    - DynDNS (static)
    - DynDNS (custom)
    - DHS
    - DyNS
    - easyDNS
    - No-IP
    - ODS.org
    - ZoneEdit
    - Loopia
    - freeDNS
    - DNSexit
    - OpenDNS
    - Namecheap.com
- More interface types (VPNs, etc) available for packet capture
- DNS Forwarder is used by the firewall itself for DNS resolution (configurable) so the firewall benefits from faster resolution via multiple concurrent queries, sees all DNS overrides/DHCP registrations, etc.
- DHCP Server can now handle arbitrary numbered options, rather than only options present in the GUI.
- Automatic update now also works for NanoBSD as well as full installs

- More configuration sections can be synchronized via XMLRPC between CARP nodes.

CHAPTER

FOUR

PRODUCT MANUALS

The pfSense Security Gateway Manuals help those who purchased appliances from Netgate get started with a new device running pfSense® software, or help get it back up and running in the case that something breaks.

Below is a list of active appliances:

- All Manuals
- Amazon AWS
- Microsoft Azure
- SG-1100
- SG-2100
- SG-3100
- SG-5100
- XG-1537
- XG-1541
- XG-7100
- XG-7100-1U

CHAPTER

FIVE

NETWORKING CONCEPTS

5.1 Understanding Public and Private IP Addresses

5.1.1 Private IP Addresses

The network standard RFC 1918 defines reserved IPv4 subnets for use only in private networks (Table *RFC 1918 Private IP Address Space*). RFC 4193 defines Unique Local Addresses (ULA) for IPv6 (Table *RFC 4193 Unique Local Address Space*). In most environments, a private IP subnet from RFC 1918 is chosen and used on all internal network devices. The devices are then connected to the Internet through a firewall or router implementing Network Address Translation (NAT) software, such as pfSense® software. IPv6 is fully routed from the internal network without NAT by Global Unicast Addresses (GUA). NAT will be explained further in *Network Address Translation*.

Table 1: RFC 1918 Private IP Address Space

CIDR Range	IP Address Range
10.0.0.0/8	10.0.0.0 - 10.255.255.255
172.16.0.0/12	172.16.0.0 - 172.31.255.255
192.168.0.0/16	192.168.0.0 - 192.168.255.255

Table 2: RFC 4193 Unique Local Address Space

Prefix	IP Address Range
fc00::/7	fc00:: - fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

A complete list of special-use IPv4 networks may be found in RFC 3330. There are private IPv4 addresses, such as 1.0.0.0/8 and 2.0.0.0/8, that have since been allocated to the dwindling IPv4 pool. Use of these addresses are problematic and not recommended. Also, avoid using 169.254.0.0/16, which according to RFC 3927 is reserved for "Link-Local" auto configuration . It should not be assigned by DHCP or set manually and routers will not allow packets from that subnet to traverse outside a specific broadcast domain. There is sufficient address space set aside by RFC 1918, so there is no need to deviate from the list shown in Table *RFC 1918 Private IP Address Space*. Improper addressing will result in network failure and should be corrected.

5.1.2 Public IP Addresses

With the exception of the largest networks, public IP addresses are assigned by Internet Service Providers. Networks requiring hundreds or thousands of public IP addresses commonly have address space assigned directly from their Regional Internet Registry (RIR). An RIR is an organization that oversees allocation and registration of public IP addresses in a designated regions of the world.

Most residential Internet connections are assigned a single public IPv4 address. Most business class connections are assigned multiple public IP addresses. A single public IP address is adequate in many circumstances and can be used in conjunction with NAT to connect hundreds of privately addressed systems to the Internet. This documentation will assist in determining the number of public IP addresses required.

Most IPv6 deployments will give the end user at least a /64 prefix network to use as a routed internal network. For each site, this is roughly 2^{64} IPv6 addresses, or 18 *quintillion* addresses, fully routed from the Internet with no need for NAT.

5.1.3 Reserved and Documentation Addresses

In addition to blocks defined in RFC 1918, RFC 5735 describes blocks reserved for other special purposes such as documentation, testing, and benchmarking. RFC 6598 updates RFC 5735 and defines address space for Carrier-grade NAT as well. These special networks include:

Table 3: RFC 5735 Reserved Address Space

CIDR Range	Purpose
192.0.2.0/24	Documentation and example code
198.51.100.0/24	Documentation and example code
203.0.113.0/24	Documentation and example code
198.18.0.0/25	Benchmarking network devices
100.64.0.0/10	Carrier-grade NAT space

The documentation uses examples with addresses from the above documentation ranges as well as RFC 1918 networks since they are more familiar to users.

Some find these addresses tempting to use for VPNs or even local networks. We cannot recommend using them for anything other than their intended purposes, but they are much less likely to be seen "in the wild" than RFC 1918 networks.

5.2 IP Subnetting Concepts

When configuring TCP/IP settings on a device, a subnet mask (Or prefix length for IPv6) must be specified. This mask enables the device to determine which IP addresses are on the local network, and which must be reached by a gateway in the device's routing table. The default LAN IP address of 192.168.1.1 with a mask of 255.255.255.0, or /24 in CIDR notation has a network address of 192.168.1.0/24. CIDR is discussed in *Understanding CIDR Subnet Mask Notation*.

5.3 IP Address, Subnet and Gateway Configuration

The TCP/IP configuration of a host consists of the address, subnet mask (or prefix length for IPv6) and gateway. The IP address combined with the subnet mask is how the host identifies which IP addresses are on its local network. Addresses outside the local network are sent to the host's configured default gateway which it assumes will pass the traffic on to the desired destination. An exception to this rule is a static route which instructs a device to contact specific non-local subnets reachable via locally connected routers. This list of gateways and static routes is kept on the routing table of each host. To see the routing table used by pfSense® software, see *Viewing Routes*. More information about routing can be found in *Routing*.

In a typical pfSense deployment, hosts are assigned an IP address, subnet mask and gateway within the LAN range of the pfSense device. The LAN IP address on pfSense becomes the default gateway. For hosts connecting by an interface other than LAN, use the appropriate configuration for the interface to which the device is connected.

Hosts within a single network communicate directly with each other without involvement from the default gateway. This means that no firewall, including pfSense, can control host-to-host communication within a network segment. If this functionality is required, hosts need to be segmented via the use of multiple switches, VLANs, or employ equivalent switch functionality like PVLAN. VLANs are covered in *Virtual LANs (VLANs)*.

5.4 Understanding CIDR Subnet Mask Notation

pfSense® firewalls use CIDR (Classless Inter-Domain Routing) notation rather than the common subnet mask 255.x.x.x when configuring addresses and networks. Refer to the *CIDR Subnet Table* to find the CIDR equivalent of a decimal subnet mask.

Table 4: CIDR Subnet Table

Subnet Mask	CIDR Prefix	Total IP Addresses	Usable IP Addresses	Number of /24 networks
255.255.255.255	/32	1	1	1/256th
255.255.255.254	/31	2	2*	1/128th
255.255.255.252	/30	4	2	1/64th
255.255.255.248	/29	8	6	1/32nd
255.255.255.240	/28	16	14	1/16th
255.255.255.224	/27	32	30	1/8th
255.255.255.192	/26	64	62	1/4th
255.255.255.128	/25	128	126	1 half
255.255.255.0	/24	256	254	1
255.255.254.0	/23	512	510	2
255.255.252.0	/22	1024	1022	4
255.255.248.0	/21	2048	2046	8
255.255.240.0	/20	4096	4094	16
255.255.224.0	/19	8192	8190	32
255.255.192.0	/18	16,384	16,382	64
255.255.128.0	/17	32,768	32,766	128
255.255.0.0	/16	65,536	65,534	256
255.254.0.0	/15	131,072	131,070	512
255.252.0.0	/14	262,144	262,142	1024
255.248.0.0	/13	524,288	524,286	2048
255.240.0.0	/12	1,048,576	1,048,574	4096
255.224.0.0	/11	2,097,152	2,097,150	8192
255.192.0.0	/10	4,194,304	4,194,302	16,384

continues on next page

Table 4 – continued from previous page

Subnet Mask	CIDR Prefix	Total IP Addresses	Usable IP Addresses	Number of /24 networks
255.128.0.0	/9	8,388,608	8,388,606	32,768
255.0.0.0	/8	16,777,216	16,777,214	65,536
254.0.0.0	/7	33,554,432	33,554,430	131,072
252.0.0.0	/6	67,108,864	67,108,862	262,144
248.0.0.0	/5	134,217,728	134,217,726	1,048,576
240.0.0.0	/4	268,435,456	268,435,454	2,097,152
224.0.0.0	/3	536,870,912	536,870,910	4,194,304
192.0.0.0	/2	1,073,741,824	1,073,741,822	8,388,608
128.0.0.0	/1	2,147,483,648	2,147,483,646	16,777,216
0.0.0.0	/0	4,294,967,296	4,294,967,294	33,554,432

Note: The use of /31 networks is a special case defined by RFC 3021 where the two IP addresses in the subnet are usable for point-to-point links to conserve IPv4 address space. Not all operating systems support RFC 3021, so use it with caution. On systems that do not support RFC 3021, the subnet is unusable because the only two addresses defined by the subnet mask are the null route and broadcast and no usable host addresses.

pfSense 2.5.2-RELEASE supports the use of /31 networks for interfaces and Virtual IP addresses.

5.4.1 Where do CIDR numbers come from?

The CIDR number comes from the number of ones in the subnet mask when converted to binary.

The common subnet mask 255.255.255.0 is 11111111.11111111.11111111.00000000 in binary. This adds up to 24 ones, or /24 (pronounced 'slash twenty four').

A subnet mask of 255.255.255.192 is 11111111.11111111.11111111.11000000 in binary, or 26 ones, hence /26.

5.5 CIDR Summarization

In addition to specifying subnet masks, CIDR can also be employed for IP or network summarization purposes. The "Total IP Addresses" column in *CIDR Subnet Table* indicates how many addresses are summarized by a given CIDR mask. For network summarization purposes, the "Number of /24 networks" column is useful. CIDR summarization can be used in several parts of the pfSense® web interface, including firewall rules, NAT, virtual IPs, IPsec, and static routes.

IP addresses or networks that can be contained within a single CIDR mask are known as "CIDR summarizable".

When designing a network, ensure all private IP subnets in use at a particular location are CIDR summarizable. For example, if three /24 subnets are required at one location, a /22 network subnetted into four /24 networks should be used. The following table shows the four /24 subnets used with the subnet 10.70.64.0/22.

Table 5: CIDR Route Summarization

10.70.64.0/22 split into /24 networks
10.70.64.0/24
10.70.65.0/24
10.70.66.0/24
10.70.67.0/24

This keeps routing more manageable for multi-site networks connected to another physical location via the use of a private WAN circuit or VPN. With CIDR summarizable subnets, one route destination covers all the networks at each location. Without it, there are several different destination networks per location.

The previous table was developed using a network calculator found at the subnetmask.info website.

The calculator converts from dotted decimal to CIDR mask, and vice versa, as shown in Figure *Subnet Mask Converter*. If the *CIDR Subnet Table* provided in this chapter is not available, this tool can be used to convert a CIDR prefix to dotted decimal notation. Enter a CIDR prefix or a dotted decimal mask and click the appropriate **Calculate** button to find the conversion.



Fig. 1: Subnet Mask Converter

Enter the dotted decimal mask into the Network/Node Calculator section along with one of the /24 networks. Click **Calculate** to populate the bottom boxes with the range covered by that particular /24 as demonstrated in Figure *Network/Node Calculator*. In this example, the network address is 10.70.64.0/22, and the usable /24 networks are 64 through 67. The term "Broadcast address" in this table refers the highest address within the range.



Fig. 2: Network/Node Calculator

5.5.1 Finding a matching CIDR network

IPv4 Ranges in the format of x.x.x.x-y.y.y.y are supported in Aliases. For Network type aliases, an IPv4 range is automatically converted to the equivalent set of CIDR blocks. For Host type aliases, a range is converted to a list of IPv4 addresses. See *Aliases* for more information.

If an exact match isn't necessary, numbers can be entered into the Network/Node Calculator to approximate the desired summarization.

5.6 Broadcast Domains

A broadcast domain is the portion of a network sharing the same layer 2 segment. Broadcast messages from hosts are sent to every port in their broadcast domain, thus hosts inside a broadcast domain can reach each other directly. For example hosts can use ARP or NDP to locate neighbors within a broadcast domain and communicate directly at layer 2 without involving an intermediate gateway router.

In a network with a single switch without VLANs, the broadcast domain is that entire switch. In a network with multiple interconnected switches without the use of VLANs, the broadcast domain includes all of those switches. When using VLANs, each VLAN is typically its own broadcast domain. The exact size of the broadcast domain in that case varies depending on how many access ports are in the VLAN, along with interconnected switches (trunked, stacked, etc).

Some switches also support special modes which segment a broadcast domain into multiple smaller isolated broadcast domains. This is sometimes called "Private VLANs", and they are typically used for security purposes. In these modes, hosts can only directly communicate between a specific set of ports, commonly limited to the host and the gateway for the segment, even if they are a part of a subnet with many other hosts. This is similar in concept to wireless AP client isolation.

Since broadcast messages are sent to every port in the broadcast domain, large broadcast domains should be avoided as they are "noisy" and do not scale well. Depending on the type of broadcast messages, some switches can optimize this behavior but it's best to plan for the worst case. For example in a network with thousands of ports on a single broadcast domain, thousands of hosts communicating among each other generate large amounts of broadcast traffic which is copied everywhere in the broadcast domain. The best practice is to keep each segment as small as possible, where feasible, to prevent switches and hosts from having to process large amounts of unnecessary broadcast traffic.

A single broadcast domain *can* contain more than one IPv4 or IPv6 subnet, however, that is generally not considered good network design. Though it appears on the surface that multiple subnets in the same broadcast domain are separate, there is no true isolation or security between them. IP subnets should be segregated into different broadcast domains via the use of separate switches or VLANs. The exception to this is running both IPv4 and IPv6 networks within a single broadcast domain. This is called dual stack and it is a common and useful technique using both IPv4 and IPv6 connectivity for hosts.

Broadcast domains can be combined by *bridging* two network interfaces together. In this scenario care must be taken to avoid switch loops where a switch ends up with a connection back to itself, creating an infinite traffic loop (*Bridging and Layer 2 Loops*). Another reason to avoid bridging is that by combining broadcast domains, both networks and the bridge between them must carry broadcast traffic for every network on the bridge. The increased load, especially for larger networks, can be significant, especially if broadcast domains are being bridged using a VPN. There are also proxies for certain protocols which do not combine broadcast domains but yield the same net effect, such as a DHCP relay which relays DHCP requests into a broadcast domain on another interface.

See also:

- *Bridging*
- *Bridging and Layer 2 Loops*
- *Virtual LANs (VLANs)*
- Broadcast Domain (Wikipedia)

5.7 IPv6

5.7.1 Basics

IPv6 allows for exponentially more IP address space than IPv4. IPv4 uses a 32- bit address, which allows for 2^{32} or over 4 billion addresses, less if the sizable reserved blocks and IPs burned by subnetting are removed. IPv6 uses a 128-bit address, which is 2^{128} or 3.403×10^{38} IP addresses. The standard size IPv6 subnet defined by the IETF is a /64, which contains 2^{64} IPs, or 18.4 *quintillion* addresses. The entire IPv4 space can fit inside a typical IPv6 subnet many times over with room to spare.

One of the more subtle improvements with IPv6 is that no IP addresses are lost to subnetting. With IPv4, two IP addresses are lost per subnet to account for a null route and broadcast IP address. In IPv6, broadcast is handled via the same mechanisms used for multicast involving special addresses sent to the entire network segment. Additional improvements include integrated packet encryption, larger potential packet sizes, and other design elements that make it easier for routers to manage IPv6 at the packet level.

Unlike IPv4, all packets are routed in IPv6 without NAT. Each IP address is directly accessible by another unless stopped by a firewall. This can be a very difficult concept to grasp for people who are used to having their LAN exist with a specific private subnet and then performing NAT to whatever the external address happens to be.

There are fundamental differences in the operation of IPv6 in comparison to IPv4, but mostly they are only that: differences. Some things are simpler than IPv4, others are slightly more complicated, but for the most part it's simply different. Major differences occur at layer 2 (ARP vs. NDP for instance) and layer 3 (IPv4 vs. IPv6 addressing). The protocols used at higher layers are identical; only the transport mechanism for those protocols has changed. HTTP is still HTTP, SMTP is still SMTP, etc.

5.7.2 Firewall and VPN Concerns

IPv6 restores true peer-to-peer connectivity originally in place with IPv4 making proper firewall controls even more important. In IPv4, NAT was misused as an additional firewall control. In IPv6, NAT is removed. Port forwards are no longer required in IPv6 so remote access will be handled by firewall rules. Care must be taken to ensure encrypted VPN LAN to LAN traffic is not routed directly to the remote site. See *IPv6 VPN and Firewall Rules* for a more in-depth discussion on IPv6 firewall concerns with respect to VPN traffic.

5.7.3 Requirements

IPv6 requires an IPv6-enabled network. IPv6 connectivity delivered directly by an ISP is ideal. Some ISPs deploy a dual stack configuration in which IPv4 and IPv6 are delivered simultaneously on the same transport. Other ISPs use tunneling or deployment types to provide IPv6 indirectly. It is also possible to use a third party provider such as Hurricane Electric's tunnelbroker service.

In addition to the service, software must also support IPv6. pfSense® has been IPv6-capable since 2.1-RELEASE. Client operating systems and applications must also support IPv6. Many common operating systems and applications support it without problems. Microsoft Windows has supported IPv6 in production-ready state since 2002 though newer versions handle it much better. OS X has supported IPv6 since 2001 with version 10.1 "PUMA". Both FreeBSD and Linux support it in the operating system. Most web browsers and mail clients support IPv6, as do recent versions of other common applications. To ensure reliability, it is always beneficial to employ the latest updates.

Some mobile operating systems have varying levels of support for IPv6. Android and iOS both support IPv6, but Android only has support for stateless auto configuration for obtaining an IP address and not DHCPv6. IPv6 is part of the LTE specifications so any mobile device supporting LTE networks supports IPv6 as well.

5.7.4 IPv6 WAN Types

Details can be found in *IPv6 Configuration Types*, but some of the most common ways of deploying IPv6 are:

Static Addressing Native and using IPv6 either on its own or in a dual stack configuration alongside IPv4.

DHCPv6 Address automatically obtained by DHCPv6 to an upstream server. Prefix delegation may also be used with DHCPv6 to deliver a routed subnet to a DHCPv6 client.

Stateless address auto configuration (SLAAC) Automatically determines the IPv6 address by consulting router advertisement messages and generating an IP address inside a prefix. This is not very useful for a router, as there is no way to route a network for the "inside" of the firewall. It may be useful for appliance modes.

6RD Tunnel A method of tunneling IPv6 traffic inside IPv4. This is used by ISPs for rapid IPv6 deployment.

6to4 Tunnel Similar to 6RD but with different mechanisms and limitations.

GIF Tunnel Not technically a direct WAN type, but commonly used. Customer builds an IPv4 GIF tunnel to a provider to tunnel IPv6 traffic.

While not technically a WAN type, IPv6 connectivity can also be arranged over OpenVPN or IPsec with IKEv2. OpenVPN and IPsec in IKEv2 mode can carry IPv4 and IPv6 traffic simultaneously, so they can deliver IPv6 over IPv4, though with more overhead than a typical tunnel broker that uses GIF. These are good options for a company that has IPv6 at a datacenter or main office but not at a remote location.

5.7.5 Address Format

An IPv6 address consists of 32 hexadecimal digits, in 8 sections of 4 digits each, separated by colons. It looks something like this: *1234:5678:90ab:cdef:1234:5678:90ab:cdef*

IPv6 addresses have several shortcuts that allow them to be compressed into smaller strings following certain rules.

If there are any leading zeroes in a section, they may be left off. *0001:0001:0001:0001:0001:0001:0001:0001* could be written as *1:1:1:1:1:1:1:1*.

Any number of address parts consisting of only zeroes may be compressed by using *::* but this can only be done once in an IPv6 address to avoid ambiguity. A good example of this is local host, compressing *0000:0000:0000:0000:0000:0000:0000:0001* to *::1*. Any time *::* appears in an IPv6 address, the values between are all zeroes. An IP address such as *fe80:1111:2222:0000:0000:0000:7777:8888*, can be represented as *fe80:1111:2222::7777:8888*. However, *fe80:1111:0000:0000:4444:0000:0000:8888* cannot be shortened using *::* more than once. It would either be *fe80:1111::4444:0:0:8888* or *fe80:1111:0:0:4444::8888* but it *cannot* be *fe80:1111::4444::8888* because there is no way to tell how many zeroes have been replaced by either *::* operator.

Determining an IPv6 Addressing Scheme

Because of the increased length of the addresses, the vast space provided in even a basic /64 subnet, and the ability to use hexadecimal digits, there is more freedom to design device network addresses.

On servers using multiple IP address aliases for virtual hosts, jails, etc, a useful addressing scheme is to use the seventh section of the IPv6 address to denote the server. Then use the eighth section for individual IPv6 aliases. This groups all of the IPs into a single recognizable host. For example, the server itself would be *2001:db8:1:1::a:1*, and then the first IP alias would be *2001:db8:1:1::a:2*, then * *2001:db8:1:1::a:3*, etc. The next server would be * *2001:db8:1:1::b:1*, and repeats the same pattern.

Some administrators like to have fun with their IPv6 addresses by using hexadecimal letters and number/letter equivalents to make words out of their IP addresses. Lists of hexadecimal words around the web can be used to create more memorable IP addresses such as *2001:db8:1:1::dead:beef*.

Decimal vs. Hexadecimal Confusion

Creating consecutive IPv6 addresses with a hexadecimal base may cause confusion. Hexadecimal values are base 16 unlike decimal values which are base 10. For example, the IPv6 address *2001:db8:1:1::9* is followed by *2001:db8:1:1::a*, *not 2001:db8:1:1::10*. By going right to *2001:db8:1:1::10*, the values a-f have been skipped, leaving a gap. Consecutive numbering schemes are not required and their use is left to the discretion of the network designer. For some, it is psychologically easier to avoid using the hexadecimal digits.

Given that all IPv4 addresses can be expressed in IPv6 format, this issue will arise when designing a dual stack network that keeps one section of the IPv6 address the same as its IPv4 counterpart.

5.7.6 IPv6 Subnetting

IPv6 subnetting is easier than IPv4. It's also different. Want to divide or combine a subnet? All that is needed is to add or chop off digits and adjust the prefix length by a multiple of four. No longer is there a need to calculate subnet start/end addresses, usable addresses, the null route, or the broadcast address.

IPv4 had a subnet mask (dotted quad notation) that was later replaced by CIDR masking. IPv6 doesn't have a subnet mask but instead calls it a Prefix Length, often shortened to "Prefix". Prefix length and CIDR masking work similarly; The prefix length denotes how many bits of the address define the network in which it exists. Most commonly the prefixes used with IPv6 are multiples of four, as seen in Table *IPv6 Subnet Table*, but they can be any number between 0 and 128.

Using prefix lengths in multiples of four makes it easier for humans to distinguish IPv6 subnets. All that is required to design a larger or smaller subnet is to adjust the prefix by multiple of four. For reference, see Table *IPv6 Subnet Table* listing the possible IPv6 addresses, as well as how many IP addresses are contained inside of each subnet.

Table 6: IPv6 Subnet Table

Prefix	Subnet Example	Total IP Addresses	# of /64 nets
4	x::	2^{124}	2^{60}
8	xx::	2^{120}	2^{56}
12	xxx::	2^{116}	2^{52}
16	xxxx::	2^{112}	2^{48}
20	xxxx:x::	2^{108}	2^{44}
24	xxxx:xx::	2^{104}	2^{40}
28	xxxx:xxx::	2^{100}	2^{36}
32	xxxx:xxxx::	2^{96}	4,294,967,296
36	xxxx:xxxx:x::	2^{92}	268,435,456
40	xxxx:xxxx:xx::	2^{88}	16,777,216
44	xxxx:xxxx:xxx::	2^{84}	1,048,576
48	xxxx:xxxx:xxxx::	2^{80}	65,536
52	xxxx:xxxx:xxxx:x::	2^{76}	4,096
56	xxxx:xxxx:xxxx:xx::	2^{72}	256
60	xxxx:xxxx:xxxx:xxx::	2^{68}	16
64	xxxx:xxxx:xxxx:xxxx::	2^{64} (18,446,744,073,709,551,616)	1
68	xxxx:xxxx:xxxx:xxxx:x::	2^{60} (1,152,921,504,606,846,976)	0
72	xxxx:xxxx:xxxx:xxxx:xx::	2^{56} (72,057,594,037,927,936)	0
76	xxxx:xxxx:xxxx:xxxx:xxx::	2^{52} (4,503,599,627,370,496)	0

continues on next page

Table 6 – continued from previous page

Prefix	Subnet Example	Total IP Addresses	# of /64 nets
80	xxxx:xxxx:xxxx:xxxx:xxxx::	2^{48} (281,474,976,710,656)	0
84	xxxx:xxxx:xxxx:xxxx:xxxx:x::	2^{44} (17,592,186,044,416)	0
88	xxxx:xxxx:xxxx:xxxx:xxxx:xx::	2^{40} (1,099,511,627,776)	0
92	xxxx:xxxx:xxxx:xxxx:xxxx:xxx::	2^{36} (68,719,476,736)	0
96	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx::	2^{32} (4,294,967,296)	0
100	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x::	2^{28} (268,435,456)	0
104	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xx::	2^{24} (16,777,216)	0
108	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxx::	2^{20} (1,048,576)	0
112	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx::	2^{16} (65,536)	0
116	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x::	2^{12} (4,096)	0
120	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xx::	2^{8} (256)	0
124	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxx::	2^{4} (16)	0
128	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx	2^{0} (1)	0

A /64 is a standard size IPv6 subnet as defined by the IETF. It is smallest subnet that can used locally if auto configuration is desired.

Typically, an ISP assigns a /64 or smaller subnet to establish service on the WAN. An additional network is routed for LAN use. The size of the allocation depends upon the ISP, but it's not uncommon to see end users receive at least a /64 and even up to a /48.

A tunnel service provider such as tunnelbroker.net run by Hurricane Electric will allocate a /48 in addition to a routed /64 subnet and a /64 interconnect.

Assignments larger than /64 usually adopt the first /64 for LAN and subdivide the rest for requirements such as VPN tunnel, DMZ, or a guest network.

5.7.7 Special IPv6 Subnets

Special use networks are reserved in IPv6. A full list of these can be found in the Wikipedia IPv6 article. Six examples of IPv6 special networks and their addresses are shown below in *IPv6 Special Networks and Addresses*.

Table 7: IPv6 Special Networks and Addresses

Network	Purpose
2001:db8::/32	Documentation prefix used for examples
::1	Localhost
fc00::/7	Unique Local Addresses (ULA) - also known as "Private" IPv6 addresses.
fe80::/10	Link Local addresses, only valid inside a single broadcast domain.
2001::/16	Global Unique Addresses (GUA) - Routable IPv6 addresses.
ff00::0/8	Multicast addresses

5.7.8 Neighbor Discovery

IPv4 hosts find each other on a local segment using ARP broadcast messages, but IPv6 hosts find each other by sending Neighbor Discovery Protocol (NDP) messages. Like ARP, NDP works inside a given broadcast domain to find other hosts inside of a specific subnet.

By sending special ICMPv6 packets to reserved multicast addresses, NDP handles the tasks of neighbor discovery, router solicitations, and route redirects similar to IPv4's ICMP redirects.

pfSense® automatically adds firewall rules on IPv6 enabled interfaces that permit NDP to function. All current known neighbors on IPv6 can viewed in the firewall GUI at **Diagnostics > NDP Table**.

5.7.9 Router Advertisements

IPv6 routers are located through their Router Advertisement (RA) messages instead of by DHCP. IPv6-enabled routers that support dynamic address assignment are expected to announce themselves on the network to all clients and respond to router solicitations. When acting as a client (WAN interfaces), pfSense accepts RA messages from upstream routers. When acting as a router, pfSense provides RA messages to clients on its internal networks. See *Router Advertisements (Or: "Where is the DHCPv6 gateway option")* for more details.

5.7.10 Address Allocation

Client addresses can be allocated by static addressing through SLAAC (*Router Advertisements (Or: "Where is the DHCPv6 gateway option")*), DHCP6 (*IPv6 Router Advertisements*), or other tunneling methods such as OpenVPN.

DHCP6 Prefix Delegation

DHCP6 Prefix Delegation delivers a routed IPv6 subnet to a DHCP6 client. A WAN- type interface can be set to receive a prefix over DHCP6 (*DHCP6*, *Track Interface*). A router functioning at the edge of a large network can provide prefix delegation to other routers inside the network (*DHCPv6 Prefix Delegation*).

5.7.11 IPv6 and NAT

Though IPv6 removes most any need for NAT, there are rare situations that call for the use of NAT with IPv6 such as Multi-WAN for IPv6 on residential or small business networks.

Gone is the traditional type of ugly port translated NAT (PAT) where internal addresses are translated using ports on a single external IP address. It is replaced by a straight network address translation called Network Prefix Translation (NPt). This is available in the pfSense® web configurator under **Firewall > NAT** on the **NPt** tab. NPt translates one prefix to another. So *2001:db8:1111:2222::/64* translates to * *2001:db8:3333:4444::/64*. Though the prefix changes, the remainder of the address will be identical for a given host on that subnet. For more on NPt, see *IPv6 Network Prefix Translation (NPt)*.

There is a mechanism built into IPv6 to access IPv4 hosts using a special address notation, such as *::ffff:192.168.1.1*. The behavior of these addresses can vary between OS and application and is unreliable.

5.7.12 IPv6 and pfSense

Unless noted otherwise, it safe to assume that IPv6 is supported by pfSense® in a given area or feature.

Some noteworthy areas of pfSense that *do not* support IPv6 are: Captive Portal and most DynDNS providers.

Note: On systems upgraded from versions of pfSense prior to 2.1, IPv6 traffic is blocked by default. To allow IPv6:

- Navigate to **System > Advanced** on the **Networking** tab
- Check **Allow IPv6**
- Click **Save**

---

pfSense Packages

Some packages are maintained by the community, so IPv6 support varies. In most cases IPv6 support depends upon the capabilities of the underlying software. It is safe to assume a package does not support IPv6 unless otherwise noted. Packages are updated periodically so it is best to test a package to determine if it supports IPv6.

5.7.13 Controlling IPv6 Preference for traffic from the firewall itself

By default, pfSense® software will prefer IPv6 when configured. If IPv6 routing is not functional but the system believes it is, pfSense may fail to check updates or download packages properly.

To change this behavior, pfSense provides a method in the GUI to control whether services on the firewall prefer IPv4 over IPv6:

- Navigate to **System > Advanced** on the **Networking** tab
- Check **Prefer to use IPv4 even if IPv6 is available**
- Click **Save**

Once the settings have been saved, the firewall itself will prefer IPv4 for outbound communication.

See also:

- *Configuring IPv6 Through A Tunnel Broker Service*

Around the world, the availability of new IPv4 addresses is declining. The amount of free space varies by region, but some have already run out of allocations and others are rapidly approaching their limits. As of January 31, 2011, IANA allocated all of its space to regional internet registries (RIRs). In turn, these RIR allocations have run out in some locations such as APNIC (Asia/Pacific), RIPE (Europe), and LACNIC (Latin America and Caribbean) for /8 networks. Though some smaller allocations are still available, it is increasingly difficult to obtain new IPv4 address space in these regions. ARIN (North America) ran out on September 24th, 2015.

To account for this, IPv6 was created as a replacement for IPv4. Available in some forms since the 1990s, factors like inertia, complexity, and the cost of developing or purchasing compatible routers and software has slowed its uptake until the last few years. Even then, it's been rather slow with only 8% of Google users having IPv6 connectivity by July 2015.

Over the years, support for IPv6 in software, operating systems, and routers has improved so the situation is primed to get better. Still it is up to ISPs to start delivering IPv6 connectivity to users. It's a catch-22 situation: Content providers are slow to provide IPv6 because few users have it. Meanwhile, users don't have it because there isn't a lot of IPv6 content and even less content available only over IPv6. Users don't know they need it so they don't demand the service from their ISPs.

Some providers are experimenting with *Carrier Grade NAT (CGN)* to stretch their IPv4 networks farther. CGN places their IPv4 residential customers behind another layer of NAT further breaking protocols that already don't deal with one layer of NAT. Mobile data providers have been doing this for some time, but the applications typically found on mobile devices aren't affected since they work as if they're behind a typical SOHO router style NAT. While solving one problem, it creates others as observed when CGN is used as a firewall's WAN, when tethering on a PC, or in some cases attempting to use a traditional IPsec VPN without NAT-T, or PPTP. ISPs employing CGN should be used only if there is no other choice.

There are many books and web sites available with volumes of in-depth information on IPv6. The Wikipedia article on IPv6, http://en.wikipedia.org/wiki/IPv6, is a great resource for additional information and links to other sources. It's worth using as a starting point for more information on IPv6. There are also many good books on IPv6 available, but be careful to purchase books with recent revisions. There have been changes to the IPv6 specification over the years and it's possible that the material could have changed since the book's printing.

See also:

pfSense Hangouts on Youtube to view the July 2015 Hangout on IPv6 Basics

This documentation is not an introduction to networks but there are certain networking concepts that need to be addressed. For readers without basic fundamental networking knowledge, we suggest locating additional introductory material as this chapter will not adequately provide all necessary information.

IPv6 concepts are introduced later in this chapter under *IPv6*. For clarity, traditional IP addresses are referred to as IPv4 addresses. Except where otherwise noted, most functions will work with either IPv4 or IPv6 addresses. The general term IP address refers to either IPv4 or IPv6.

5.8 Brief introduction to OSI Model Layers

The OSI model has a network framework consisting of seven layers. These layers are listed in hierarchy from lowest to highest. A brief overview of each level is outlined below. More information can be found in many networking texts and on Wikipedia (http://en.wikipedia.org/wiki/OSI_model).

Layer 1 - Physical Refers to either electrical or optical cabling that transports raw data to all the higher layers.

Layer 2 - Data Link Typically refers to Ethernet or another similar protocol that is being spoken on the wire. This documentation often refers to layer 2 as meaning the Ethernet switches or other related topics such as ARP and MAC addresses.

Layer 3 - Network Layer The protocols used to move data along a path from one host to another, such as IPv4, IPv6, routing, subnets etc.

Layer 4 - Transport Layer Data transfer between users, typically refers to TCP or UDP or other similar protocols.

Layer 5 - Session Layer Manages connections and sessions (typically referred to as "dialogs") between users, and how they connect and disconnect gracefully.

Layer 6 - Presentation Layer Handles any conversions between data formats required by users such as different character sets, encodings, compression, encryption, etc.

Layer 7 - Application Layer Interacts with the user or software application, includes familiar protocols such as HTTP, SMTP, SIP, etc.

CHAPTER

SIX

HARDWARE

6.1 Minimum Hardware Requirements

The minimum hardware requirements for pfSense® 2.5.2-RELEASE on hardware not sold by Netgate are:

- 64-bit amd64 (x86-64) compatible CPU
- 1GB or more RAM
- 8 GB or larger disk drive (SSD, HDD, etc)
- One or more compatible network interface cards
- Bootable USB drive or high capacity optical drive (DVD or BD) for initial installation

Note: The minimum requirements are not suitable for all environments; see *Hardware Sizing Guidance* for details.

6.2 Hardware Selection

The use of open source operating systems with untested hardware may create hardware/software conflicts. *Hardware Tuning and Troubleshooting* offers tips on resolving various issues.

6.2.1 Preventing hardware headaches

Use Genuine Netgate Hardware

The best practice is to use hardware from the Netgate Store. Netgate hardware has been developed to assure that specific hardware platforms have been thoroughly tested and validated.

Search for the experiences of others

The experiences of others are a valuable source of knowledge which can be found by researching pfSense software and hardware compatibility online, especially on the Netgate Forum. Reports of failure are not necessarily considered definitive because problems can arise from a number of issues other than hardware incompatibility.

If the hardware in question is from a major manufacturer, an internet search by make, model, and `site:netgate.com` will search the Netgate website for relevant user experiences. Searching for the make, model, and `pfSense` will find user experiences on other websites. Repeating the same search with `FreeBSD` instead of `pfSense` can also turn up useful experiences.

Naming Conventions

This documentation refers to the 64-bit hardware architecture as `amd64`, the architecture designation used by FreeBSD. Intel adopted the architecture created by AMD for x86-64, thus the name amd64 refers to all x86 64-bit CPUs.

Netgate sells ARM appliances compatible with its factory edition of pfSense software. This hardware is based on the `armv6` and `armv7` architectures (also called `arm`) and `aarch64` (also called `arm64`). Items specific to those unique architectures will be called out as necessary. The generic term ARM may be considered to apply to all of these, but **only** for the specific ARM-based appliances sold by Netgate, such as the 2100 and 3100.

6.3 Hardware Sizing Guidance

When sizing hardware for pfSense® software, required throughput and necessary features are the primary factors that govern hardware selection.

The information on Netgate Store now contains up-to-date specifications and performance data on all hardware sold by Netgate. The data on the Netgate Store is updated as needed and it is always the most accurate and current source of performance data.

Tip: Contact Netgate Sales for personalized help in selecting the most suitable model for any implementation.

Estimating throughput of third party / whitebox hardware is difficult and inaccurate. In some cases, ballpark estimates may be made by comparing hardware specifications with those found on the Netgate Store for comparable models.

6.3.1 Throughput Considerations

In real networks the traffic flow will likely contain packets of varying size, not all maximum size packets, but it completely depends on the environment and the type of traffic involved. IMIX testing attempts to approximate a mixture of traffic that more closely resembles real-world environments. Simple IMIX traffic is sets of 7 (40) byte packets, (4) 576 byte packets, 1 (1500) byte packets, plus Ethernet framing overhead.

Note: The Netgate Store entries for hardware include data for both maximum size packet size ("IPERF3") as well as results for IMIX traffic patterns.

As a general reference, table *500,000 PPS Throughput at Various Frame Sizes* lists a few common packet sizes and the throughput achieved at an example rate of 500,000 packets per second.

Table 1: 500,000 PPS Throughput at Various Frame Sizes

Frame size	Throughput at 500 Kpps
64 bytes	244 Mbps
500 bytes	1.87 Gbps
1000 bytes	3.73 Gbps
1500 bytes	5.59 Gbps

Performance difference by network adapter type

The choice of NIC has a significant impact on performance. Inexpensive, low end cards consume significantly more CPU than better quality cards such as Intel. The first bottleneck with firewall throughput is the CPU. Throughput improves significantly by using a better quality NIC with slower CPUs. By contrast, increasing the speed of the CPU will not proportionally increase the throughput when coupled with a low quality NIC.

6.3.2 Feature Considerations

Features, services and packages enabled on the firewall can lower the total potential throughput as they consume hardware resources that could otherwise be used to transfer network traffic. This is especially true for packages that intercept or inspect network traffic, such as Snort or Suricata.

Most base system features do not significantly factor into hardware sizing but a few can potentially have a considerable impact on hardware utilization.

Large State Tables

Active network connections through the firewall are tracked in the firewall state table. Each connection through the firewall consumes two states: One entering the firewall and one leaving the firewall. For example, if a firewall must handle 100,000 simultaneous web server client connections the state table must be able to hold 200,000 states.

See also:

States are covered further in *Firewall*.

Firewalls in environments which require large numbers of simultaneous states must have sufficient RAM to contain the state table. Each state takes approximately 1 KB of RAM, which makes calculating the memory requirements relatively easy. Table *Large State Table RAM Consumption* provides a guideline for the amount of memory required for larger state table sizes. This is solely the memory used for the state tracking. The operating system itself along with other services will require at least 175-256 MB additional RAM and possibly more depending on the features used.

Table 2: Large State Table RAM Consumption

States	Connections	RAM Required
100,000	50,000	~97 MB
500,000	250,000	~488 MB
1,000,000	500,000	~976 MB
3,000,000	1,500,000	~2900 MB
8,000,000	4,000,000	~7800 MB

It is safer to overestimate the requirements. Based on the information above, a good estimate would be that 100,000 states consume about 100 MB of RAM, or that 1,000,000 states would consume about 1 GB of RAM.

VPN (all types)

The question customers typically ask about VPNs is "How many connections can my hardware handle?" That is a secondary factor in most deployments and is of lesser consideration. That metric is a relic of how other vendors have licensed VPN capabilities in the past and has no specific direct equivalent in pfSense software. The primary consideration in hardware sizing for VPN is the potential throughput of VPN traffic.

Encrypting and decrypting network traffic with all types of VPNs is CPU intensive. pfSense offers several cipher options for use with IPsec. The various ciphers perform differently and the maximum throughput of a firewall is dependent on the cipher used and whether or not that cipher can be accelerated by the hardware.

See also:

The Netgate Store contains VPN performance data for each device sold by Netgate using the most optimal cipher for each device based on its capabilities.

Hardware *cryptographic accelerators*, such as those found on most Netgate hardware, greatly increase maximum VPN throughput and largely eliminate the performance difference between accelerated ciphers. For IPsec, ciphers may be accelerated by onboard cryptographic accelerators. For example, AES-GCM is accelerated by AES-NI and it is faster not only for that, but because it also does not require a separate authentication algorithm. IPsec also has less per-packet operating system processing overhead than OpenVPN, so for the time being IPsec will nearly always be faster than OpenVPN.

Where high VPN throughput is a requirement for a firewall, hardware cryptographic acceleration is of utmost importance to ensure not only fast transmission speeds but also reduced CPU overhead. The reduction in CPU overhead means the VPN will not lower the performance of other services on the firewall.

The current best available acceleration is available by using pfSense Plus on hardware with a QAT device, or failing that, a CPU which includes AES-NI support combined with AES-GCM in IPsec.

Packages

Certain packages have a significant impact on hardware requirements, and their use must be taken into consideration when selecting hardware.

Snort/Suricata

Snort and Suricata are pfSense packages for network intrusion detection. Depending on their configuration, they can require a significant amount of RAM. 1 GB should be considered a minimum but some configurations may need 2 GB or more, not counting RAM used by the operating system, firewall states, and other packages.

Suricata is multi-threaded and can potentially take advantage of NETMAP for inline IPS if the hardware offers support.

6.4 Hardware Tuning and Troubleshooting

The underlying operating system beneath pfSense® software can be fine-tuned in several ways. A few of these tunables are available under **Advanced Options** (See *System Tunables Tab*). Others are outlined in the FreeBSD main page tuning(7).

The default installation includes a well-rounded set of values tuned for good performance without being overly aggressive. There are cases where hardware or drivers necessitate changing values or a specific network workload requires changes to perform optimally.

The hardware sold in the Netgate Store is tuned further since Netgate has detailed knowledge of the hardware, removing the need to rely on more general assumptions.

Note: Changes in `/boot/loader.conf.local` require a firewall reboot to take effect.

- *General Issues*
  - *Mbuf Exhaustion*
  - *NIC Queue Count*
  - *Disable MSIX*
  - *PPPoE with Multi-Queue NICs*
  - *TSO/LRO*
  - *IP Input Queue (intr_queue)*
- *Card-Specific Issues*
  - *Broadcom bce(4) Cards*
  - *Broadcom bge(4) Cards*
  - *Chelsio cxgbe(4) Cards*
  - *Intel igb(4) and em(4) Cards*
  - *Intel ix(4) Cards*
  - *VMware vmx(4) Interfaces*
  - *Flow Control*

6.4.1 General Issues

Mbuf Exhaustion

A common problem encountered by users of commodity hardware is mbuf exhaustion. To oversimplify, "mbufs" are network memory buffers; portions of RAM set aside for use by networking for moving data around.

The count of active mbufs is shown on the dashboard and is tracked by a graph under **Status > Monitoring**.

See also:

For details on mbufs and monitoring mbuf usage, see *Mbuf Clusters*.

If the firewall runs out of mbufs, it can lead to a kernel panic and reboot under certain network loads that exhaust all available network memory buffers. In certain cases this condition can also result in expected interfaces not being initialized and made available by the operating system. This is more common with NICs that use multiple queues or are otherwise optimized for performance over resource usage.

Additionally, mbuf usage increases when the firewall is using certain features such as *Limiters*.

To increase the amount of mbufs available, add the following to `/boot/loader.conf.local`:

```
kern.ipc.nmbclusters="1000000"
```

That number can be again be doubled or more as needed, but be careful not to exceed available kernel memory. On 64 bit systems with multiple GB of RAM, set it to 1 million (1000000).

Some network interfaces may need other similar values raised such as `kern.ipc.nmbjumbop`. In addition to the graphs mentioned above, check the output of the command `netstat -m` to verify if any areas are near exhaustion.

NIC Queue Count

For performance reasons some networks cards use multiple queues for processing packets. On multi-core systems, usually a driver will want to use one queue per CPU core. A few cases exist where this can lead to stability problems, which can be resolved by reducing the number of queues used by the NIC. To reduce the number of queues, specify the new value in `/boot/loader.conf.local`, such as:

```
hw.igb.num_queues=1
```

The name of the sysctl OID varies by network card, but it is usually located in the output of `sysctl -a`, under `hw.<drivername>`.

Disable MSIX

Message Signaled Interrupts are an alternative to classic style Interrupts for retrieving data from hardware. Some cards behave better with MSI, MSIX, or classic style Interrupts, but the card will try the best available choice (MSIX, then MSI, then Interrupts).

MSIX and MSI can be disabled via loader tunables. Add the following to `/boot/loader.conf.local`:

```
hw.pci.enable_msix=0
hw.pci.enable_msi=0
```

To nudge the card to use MSI, disable only MSIX. To nudge the card to use regular Interrupts, disable both MSI and MSIX.

PPPoE with Multi-Queue NICs

Network cards which support multiple queues rely on hashing to assign traffic to a particular queue. This works well with IPv4/IPv6 TCP and UDP traffic, for example, but fails with other protocols such as those used for PPPoE.

This can lead to a network card under performing with the default network settings, as noted on #4821 and FreeBSD PR 203856. This problem primarily affects systems with multiple CPUs and/or CPU cores, as those are the systems which benefit most from multiple NIC queues.

Adding a **System Tunable** or `loader.conf.local` entry for `net.isr.dispatch=deferred` can lead to performance gains on affected hardware.

Tuning the values of `net.isr.maxthreads` and `net.isr.numthreads` may yield additional performance gains. Generally these are best left at default values matching the number of CPU cores, but depending on the workload may work better at lower values.

Warning: In the past, `deferred` mode has led to issues on 32-bit platforms, such as crashes/panics, especially with ALTQ. There have been no recent reports, however, so it should be safe on current releases.

TSO/LRO

The settings for **Hardware TCP Segmentation Offload** (TSO) and **Hardware Large Receive Offload** (LRO) under **System > Advanced** on the **Networking** tab default to **checked** (disabled) for good reason. Nearly all hardware/drivers have issues with these settings, and they can lead to throughput issues. Ensure the options are checked. Sometimes disabling via sysctl is also necessary.

Add the following to `/boot/loader.conf.local`:

```
net.inet.tcp.tso=0
```

IP Input Queue (intr_queue)

This will show the current setting:

```
sysctl net.inet.ip.intr_queue_maxlen
```

However, in largely loaded installations this may not be enough. Here is how to check:

```
sysctl net.inet.ip.intr_queue_drops
```

If the above shows values above `0`, try doubling the current value of `net.inet.ip.intr_queue_maxlen`.

For example:

```
sysctl net.inet.ip.intr_queue_maxlen=3000
```

Keep performing the above until the point is found where drops are eliminated without any adverse effects.

Afterwards, add an entry under **System > Advanced**, **System Tunables** tab to set `net.inet.ip.intr_queue_maxlen` to `3000`

6.4.2 Card-Specific Issues

Broadcom bce(4) Cards

Several users have noted issues with certain Broadcom network cards, especially those built into Dell hardware. If `bce` interfaces are behaving erratically, dropping packets, or causing crashes, then the following tweaks may help.

Add the following to `/boot/loader.conf.local`:

```
kern.ipc.nmbclusters="1000000"
hw.bce.tso_enable=0
hw.pci.enable_msix=0
```

That will increase the amount of network memory buffers, disable TSO directly, and disable msix.

Packet loss with many (small) UDP packets

If a lot of packet loss is observed with UDP on bce cards, try changing the `netisr` settings. These can be set as system tunables under **System > Advanced**, on the **System Tunables** tab. On that page, add two new tunables:

```
net.isr.direct_force=1
net.isr.direct=1
```

Broadcom bge(4) Cards

See above, but change "bce" to "bge" in the setting names.

Chelsio cxgbe(4) Cards

It is possible to disable the allocation of resources that are not related to the router so that the network adapter can use its entire set of resources for the corresponding functions:

Add the following to `/boot/loader.conf.local`:

```
hw.cxgbe.toecaps_allowed=0
hw.cxgbe.rdmacaps_allowed=0
hw.cxgbe.iscsicaps_allowed=0
hw.cxgbe.fcoecaps_allowed=0
```

Intel igb(4) and em(4) Cards

Certain intel igb cards, especially multi-port cards, can easily exhaust mbufs and cause kernel panics. The following tweak will prevent this from being an issue. Add the following to `/boot/loader.conf.local`:

```
kern.ipc.nmbclusters="1000000"
```

That will increase the amount of network memory buffers, allowing the driver enough headroom for its optimal operation.

Intel ix(4) Cards

In `/boot/loader.conf.local`:

```
kern.ipc.nmbclusters="1000000"
kern.ipc.nmbjumbop="524288"
```

As a sysctl (system tunable):

```
hw.intr_storm_threshold=10000
```

VMware vmx(4) Interfaces

VMware VMXNET interfaces support multiple queues but do not utilize them by default. Multiple queues enable network performance to scale with the number of vCPUs and allows for parallel packet processing.

The following example values are for a virtual machine with 8 vCPUs.

Edit or create `/boot/loader.conf.local` and add the following content:

```
hw.pci.honor_msi_blacklist=0
hw.vmx.txnqueue=8
hw.vmx.rxnqueue=8
hw.vmx.txndesc=2048
hw.vmx.rxndesc=2048
```

Save the file, then reboot and check the queues with `vmstat -i` at a command prompt.

Flow Control

In some circumstances, flow control may need to be disabled. The exact method depends on the hardware involved, as in the following examples:

These example entries go in `/boot/loader.conf.local`:

cxgbe(4)

```
hw.cxgbe.pause_settings=0
```

These example entries go in **System > Advanced**, on the **System Tunables** tab (*System Tunables Tab*):

ixgbe(4) (aka ix)

```
dev.ix.x.fc=0
```

Note: One line is required for each network interface present on the system, with `x` replaced by a device id such as `0`, `1`, etc.

igc(4)

```
dev.igc.x.fc=0
```

Note: One line is required for each network interface present on the system, with `x` replaced by a device id such as `0`, `1`, etc.

igb(4)

```
dev.igb.x.fc=0
```

Note: One line is required for each network interface present on the system, with `x` replaced by a device id such as `0`, `1`, etc.

em(4)

```
dev.em.x.fc=0
```

Note: One line is required for each network interface present on the system, with `x` replaced by a device id such as `0`, `1`, etc.

For ix and others, the flow control value can be further tuned:

0 No Flow Control

1 Receive Pause

2 Transmit Pause

3 Full Flow Control, Default

6.5 Console Types

There are two console types available with pfSense® software, VGA and Serial. The active default console depends on the image/installer used and configuration settings. The difference between the two console types is explained in more detail below.

6.5.1 VGA Console

The VGA (video) console is a console with a monitor and keyboard. The video console requires hardware with a connection for a monitor (e.g. HDMI, VGA) and keyboard (USB, PS/2). In some cases a serial BIOS that does VGA redirection may work.

The VGA console is active by default using the normal memstick installer or ISO.

6.5.2 Serial Console

The serial console uses a serial/COM port to communicate with a serial client. It is primarily intended for systems without a monitor or keyboard. The serial console can also be used on systems where those are either not available or not wanted, so long as the hardware has at attached (non-USB) serial port.

The serial console is active by default when installing using the *serial* memstick and may be enabled under **System > Advanced** on VGA images.

Accessing the serial console requires a null modem serial cable attached between the `COM1` port on the firewall and a serial client. A hardware serial port is required on the firewall, but the client may use a USB serial adapter if needed.

Serial clients are quite common, often pre-installed on an operating system or easily available. The free PuTTY client is the most popular GUI choice. Other choices include GNU `screen`, `tip`, `cu` and `minicom`.

See also:

See *Connect to the Console* for details on how to connect to a serial console.

The default speed of the serial port is `115200/8/N/1`. The serial port speed may be changed under **System > Advanced**.

If the device has a BIOS accessible over serial console, it is also possible that it will not be using the same serial speed that the OS is using.

The most common serial speeds to try would be: `115200`, `38400`, and `9600`.

If the BIOS serial speed does not match the OS serial speed, the best practice is to adjust one or the other to match, so that POST messages may be viewed as well as the OS messages without having to adjust the client

6.6 Connect to the Console

A connection to the console on the target hardware is a requirement to run the installer.

For hardware with a VGA console, this is as simple as connecting a monitor and keyboard.

For hardware with a serial console, the process is more involved and requires a client PC with an appropriate port and terminal software. Follow the instructions below to connect using a serial console.

6.6.1 Connecting to a Serial Console

The instructions in this section cover general serial console topics. Some devices, such as firewalls from the Netgate Store, require slightly different methods to connect to the serial console. For devices from the Netgate Store, visit the Netgate Documentation for model-specific serial console instructions.

Serial Console Requirements

Connecting to a serial console on most firewalls requires the correct hardware on every part of the link, including:

- The client PC must have a physical serial port or a USB-to-Serial adapter
- The firewall must have a physical serial port
- A null modem serial cable and/or adapter, or a device-specific serial cable
- A terminal program on the client, such as PuTTY
- The correct serial settings for the client software

For most of the firewalls purchased from the Netgate Store, the only hardware requirement is a USB A to Mini-B cable. See Netgate Documentation for specifics.

In addition to the proper hardware connection, a serial console client program must also be available on the client PC, and the serial speed and other settings must be available.

Locating a Serial Port (Server/Firewall)

First, ensure the firewall hardware has a serial port. To use the serial console, the hardware must have a physical serial port at `COM1`. Embedded units typically have a DB9 (9-pin) serial port, but some have an RJ45 style console connector with an adapter cable that ends with a DB9 connector.

Connect a Serial Cable

First, a null modem serial cable must be connected between the firewall and a client PC. Depending on the serial port and cable being used, a serial cable gender changer may also be necessary to match the available ports.

If a real null modem serial cable is unavailable, a null modem adapter can be used to convert a standard serial cable into a null modem cable.

If the client PC does not have a physical serial port, use a USB-to-Serial adapter.

Locate the Client Serial Port

On the client PC, the serial port device name must be determined so that the client software can be used on the correct port.

Windows

On Windows clients, a physical serial port is typically `COM1`. With a USB-to-Serial adapter, it may be `COM3`. Open **Device Manager** in Windows and expand **Ports (COM & LPT)** to find the port assignment.

Mac OS X

On Mac OS X, the name can be tricky for a user to determine since it varies based on the driver name and type. Some common examples include `/dev/cu.SLAB_USBtoUART` and `/dev/cu.usbserial-<model>`.

Linux

The device associated with a USB-to-Serial adapter is likely to show up as `/dev/ttyUSB0`. Look for messages about the device attaching in the system log files or by running `dmesg`.

Note: If the device does not appear in `/dev/`, check to see if the device requires additional drivers.

FreeBSD

The device associated with a USB-to-Serial adapter is likely to show up as `/dev/cuaU0`. Look for messages about the device attaching in the system log files or by running `dmesg`.

Determine Serial Console Settings

The settings for the serial port, including the speed, must be known before a client can successfully connect to a serial console.

Whichever serial client is used, ensure that it is set for the proper Speed (`115200`), Data Bits (`8`), Parity (`No`), and Stop Bits (`1`). This is typically written as `115200/8/N/1`.

Note: Some hardware defaults to a slower speed. This is relevant to the BIOS and initial output, not to pfSense® which defaults to `115200`.

Many serial clients default to `9600/8/N/1`, so adjusting these settings is required to connect. Use `115200/8/N/1` with pfSense regardless of the setting of the hardware/BIOS.

For hardware using BIOS serial speeds other than `115200`, change the baud rate to `115200` in the BIOS setup so the BIOS and pfSense are both accessible with the same settings. Refer to the hardware manual for information on setting its baud rate.

`115200` is the default speed pfSense uses out of the box, but the serial speed used by pfSense can be changed later. See *Serial Console Speed*.

Locate a Serial Client

A serial client program must be used on the client PC. The most popular client for Windows is PuTTY, which is free and works well. PuTTY is also available for Linux and can be installed on OS X using brew. On UNIX and UNIX-Like operating systems, the `screen` program is readily available or easily installed and it can also be used to connect to serial ports from a terminal program or system console.

Windows

PuTTY is the most popular free choice for serial communication on Windows. SecureCRT is another client that works well.

Warning: Do not use Hyperterminal. Even if it is already present on the client PC, it is unreliable and prone to formatting incorrectly and losing data.

Mac OS X

On Mac OS X clients, the GNU `screen` utility is the easiest and most common choice. ZTerm and `cu` (similar to FreeBSD) can be used as well.

Linux

On Linux clients, the GNU `screen` utility is the easiest and most common choice. Programs such as PuTTY, `minicom`, or `dterm` can be used as well.

FreeBSD

On FreeBSD clients, the GNU `screen` utility is the easiest and most common choice.

As an alternative, use the built-in program `tip`. Typing `tip com1` (Or `tip ucom1` if using a USB serial adapter) will connect to the first serial port. Disconnect by typing `~.` at the start of a line.

Start a Serial Client

Now that all of the requirements have been met, it is time to run the serial client.

If the client software is not covered in this section, consult its documentation to determine how to make a serial connection.

PuTTY

- Start PuTTY
- Select *Serial* for the **Connection Type**
- Enter the serial port device name for **Serial Line**, e.g. `COM3` or `/dev/ttyUSB0`.
- Enter the appropriate **Speed**, e.g. `115200`
- Click **Open**

MINICOM

```
$ minicom -D /dev/ttyUSB0 -R 115200
```

GNU screen

- Open a terminal / command prompt
- Invoke the `screen` command using the path to the serial port, for example:

```
$ sudo screen /dev/ttyUSB0 115200
```

In some cases there may be a terminal encoding mismatch. If this happens, run `screen` in UTF-8 mode:

```
$ sudo screen -U /dev/cu.SLAB_USBtoUART 115200
```

The standard screen controls apply. Press `Ctrl-A`, `\` to quit, or `Ctrl-A`, `Ctrl-\` in some cases.

tip

The `tip` command on FreeBSD consults `/etc/remotes` and connects to serial ports based on the settings there. To setup a connection to a USB-to-serial adapter at `115200`, add a line such as the following to `/etc/remote`:

```
ucom1fast:dv=/dev/cuaU0:br#115200:pa=none:
```

To access the port, invoke `tip`:

```
$ tip ucom1fast
```

To quit, press **Enter**, then type `~.`. If connected through a terminal ssh client, `~~.` may need to be used instead so that the ssh client itself doesn't interpret the keys.

6.7 Cryptographic Accelerator Support

Cryptographic acceleration is available on some platforms, typically on hardware that has it available in the CPU like AES-NI, or built into the board such as the ones used on Netgate ARM-based systems. Most cryptographic accelerator hardware supported by FreeBSD will work, provided the drivers are in the kernel or available as loadable modules.

Note: Some modules and hardware are only supported by pfSense Plus software.

6.7.1 Supported Devices

Currently supported cryptographic accelerator devices include:

AES-NI Supported natively by most modern CPUs.

Intel QuickAssist Technology (QAT) [Plus only] Supported on certain Intel-based platforms such as select models of c3000 and c2000 SoCs, and also by QAT add-on cards. Present on several Netgate hardware models such as the 7100, 6100, 5100, and more.

CESA [Plus only] Present on some ARM platforms such as the Netgate 3100.

SafeXcel [Plus only] Present on some ARM platforms such as the Netgate 2100 and 1100.

Note: For specifics on which hardware accelerators are available on Netgate hardware, and relevant performance data, visit the Netgate Store.

6.7.2 Activating the Hardware

Some hardware acceleration is active at all times and there is no way to disable it short of removing the crypto card if it is a hardware add-on. For example, CESA acceleration cannot be disabled because it's an integrated feature of the system and the drivers are present the kernel.

Others, such as QAT, AES-NI, or SafeXcel require choosing the appropriate module under **System > Advanced** on the **Miscellaneous** tab. Choose the appropriate module to match the hardware for **Cryptographic Hardware** and then Save. The module will be loaded and available immediately.

To deactivate a loaded module, select *None* for **Cryptographic Hardware**, Save, and then reboot the system.

6.7.3 Confirming Accelerator Use

Confirming that the cryptographic acceleration device is being used by the firewall can be tricky, depending on the hardware in question.

Most often the evidence of cryptographic accelerator use is apparent in one or more of the following observations:

- Increased VPN throughput
- Decreased system load (e.g. CPU utilization) for similar levels of VPN throughput

In cases where it is not clear, some cryptographic accelerators show signs of use by checking for interrupt activity on the device using `vmstat -i | grep <name>`, where `<name>` corresponds to the name of the device:

QAT Use the shell command `vmstat -i | grep qat`

CESA Use the shell command `vmstat -i | grep cesa`

SafeXcel Use the shell command `vmstat -i | grep safexcel`

In each of these cases, first check that there is any output at all. If the device has not been used at all since the firewall last rebooted or loaded the device driver, there will be no output from the command.

Note: To see if the driver is loaded, check `kldstat -v | grep <name>` to ensure the driver is present, and check `dmesg | grep <name>` to see if the device was detected.

If there is output from `vmstat -i` for the device, check the third entry on the line, which is the total number of interrupts observed on the device(s). If this number is increasing with VPN activity, the device is being used by the firewall. For example:

```
# vmstat -i | grep qat
irq300: qat0                       5481147          3
```

In that output the `5481147` number represents the number of interrupts on the `qat0` device. Run the command again after transferring data across the VPN, and compare the number.

Note: If the command produces no output at all, the device is not being used or the device driver is not loaded.

6.7.4 Verifying Cipher Support

To see a list of engines and associated transforms supported by the hardware and active modules though OpenSSL, run:

```
/usr/bin/openssl engine -t -c
```

Note: That is only for support via OpenSSL. Other areas such as IPsec may support additional methods not listed.

6.7.5 Practical Use

IPsec

IPsec will take advantage of acceleration automatically when an active accelerator supports the cipher chosen for a tunnel. For QAT and AES-NI, the optimal cipher choice is AES-GCM.

OpenVPN

To take advantage of acceleration in OpenVPN, choose a supported cipher on each end of a given tunnel.

Nothing needs selected for OpenVPN to utilize AES-NI. The OpenSSL engine has its own code for handling AES-NI that works well without using additional modules.

6.8 Disabling Sounds/Beeps

Some hardware has a PC Speaker which can be used as a means of notification. By default, the firewall will play a tone at startup/shutdown and will emit a beep when a user logs into the GUI. Additionally, some packages are capable of producing beeps for events.

6.8.1 Disable Startup/Shutdown Tune

The startup and shutdown tunes may be disabled as follows:

- Navigate to **System > Advanced**, **Notifications** tab
- Check **Disable the startup/shutdown beep**
- Click **Save**

6.8.2 Disable Login Beep

The GUI login beep happens because the GUI login event is recorded by syslog under the LOG_AUTH facility. Messages in this facility trigger the operating system to generate a beep. To disable the beep, the GUI login messages must be suppressed as follows:

- Navigate to **System > Advanced**, **Admin Access** tab
- Check **Disable logging of webConfigurator successful logins**
- Click **Save**

6.8.3 Disable All Sounds

As an alternative, the system bell may be disabled globally:

- Navigate to **System > Advanced**, **System Tunables** tab
- Click  to create a new tunable entry using the following values:

  Tunable `kern.vt.enable_bell`

  Description `Control system sounds`

  Value `0`
- Click **Save**

See also:

- *Halting and Powering Off the Firewall*
- *Rebooting the Firewall*
- *Network Interface Drivers with ALTQ Traffic Shaping Support*
- *Troubleshooting Disk and Filesystem Issues*
- *Troubleshooting Boot Issues*
- *Troubleshooting DMA and LBA Errors*
- *Troubleshooting High CPU Load*

- *Troubleshooting Disk and Filesystem Issues*
- *Troubleshooting Lost Traffic or Disappearing Packets*
- *Troubleshooting Unexpected Reboots*

The pfSense® software distribution is compatible with most hardware supported by FreeBSD.

Current versions of pfSense software are compatible with 64-bit (amd64, x86-64) architecture hardware and Netgate ARM-based firewalls.

Alternate hardware architectures such as Raspberry Pi, other Non-Netgate ARM devices, PowerPC, MIPS, SPARC, etc. are not supported.

6.9 Hardware Compatibility

The best way to ensure that hardware is compatible with pfSense software is to buy hardware from the Netgate Store that has been tested and known to work well with pfSense software. The hardware in the store is tested with each release of pfSense software and is tuned for optimal performance.

For home-built solutions, the FreeBSD Hardware Notes for the FreeBSD version used in a given build of pfSense software is the best resource for determining hardware compatibility. pfSense software version 2.5.2-RELEASE is based on 12.2-STABLE@f4d0bc6aa6b. Another good resource is the Hardware section of the FreeBSD FAQ.

6.9.1 Network Adapters

A wide variety of wired Ethernet Network Interface Cards (NICs) are supported by FreeBSD, and are thus compatible with pfSense software. However, not all NICs are created equal. The hardware can vary greatly in quality from one manufacturer to another.

The best practice is to use Intel NICs because they have solid driver support in FreeBSD and they perform well. Most hardware sold in the Netgate Store contains Intel NICs.

Of the various other PCIe/PCI cards supported by FreeBSD, some work fine, others may suffer from instability or poor performance. In some cases, FreeBSD may support a particular NIC but specific implementations of the chipset may be lower in quality or have have poor driver support. When in doubt, search the Netgate Forum for experiences of others using the same or similar hardware.

When a firewall requires the use of VLANs, select adapters that support VLAN processing in hardware. This is discussed in *Virtual LANs (VLANs)*.

USB Network Adapters

USB network adapters of any make/model should not be used due to their unreliability and poor performance.

Wireless Adapters

Supported wireless adapters and recommendations are covered in *Wireless*.

CHAPTER

SEVEN

INSTALLING AND UPGRADING

Hardware from the Netgate Store is pre-loaded with pfSense® software. To reinstall pfSense software or to install it to other hardware, download an installer image as described in this chapter.

Warning: Hardware pre-loaded with pfSense software from commercial vendors other than the Netgate Store or authorized partners must not be trusted. Third parties may have made unauthorized, unknown alterations or additions to the software. Selling pre-loaded copies of pfSense software is a violation of the Trademark Usage Guidelines.

If pfSense software was pre-loaded on third party hardware by a vendor, wipe the system and reinstall it with a genuine copy.

If something goes wrong during the installation process, see *Troubleshooting Installation Issues*.

This chapter also covers upgrading pfSense software installations (*Upgrade Guide*) which keeps them up-to-date with the latest security, bug fixes, and new features.

7.1 Download Installation Media

Note: Customers who have purchased firewalls from the Netgate Store can get factory installation images by contacting support. The Netgate Product Manuals contain specific instructions for each model. Some Netgate devices can also run Community Edition images, but the factory images offer the best user experience.

For other hardware, continue reading.

- Navigate to the download page on pfsense.org in a web browser on a client PC.
- Select an **Architecture**:

  AMD64 (64-bit) For 64-bit x86-64 Intel or AMD hardware.

  Netgate ADI For some `amd64` SG-series firewalls from the Netgate Store, specifically, models which contain a USB console port on `COM2`.

- Select a **Platform**:

  USB Memstick Installer A disk image which can be written to a USB memory stick (memstick) and booted on the target hardware for installation.

  DVD Image (ISO) Installer To install from optical media or for use with IPMI or hypervisors which can boot from ISO images.

- Select a **Console** for *USB Memstick Installer* images:

VGA Installs using a monitor and keyboard connected to the target hardware, or virtual machines with equivalent components.

Serial Installs using a serial console on `COM1` of the target hardware. This option requires a non-USB hardware console port.

Note: Some hardware contains a usable serial port which is exposed through a special internal USB/Serial connection and dedicated USB console port. This hardware generally works fine, and is not the same as a USB/Serial adapter plugged into a USB port, which will not work for serving a serial console.

- Select a **Mirror** that is close to the client PC geographically.
- Click  **Download**.
- Copy or download the SHA-256 sum displayed by the page to *verify the download*.

Tip: To view a listing of all files on the mirror, do not select any options from the drop-down menus except for a mirror then click **Download**. A description of the file names are available on the downloads page.

See also:

At any point in the installation if something does not go as described, check *Troubleshooting Installation Issues*.

7.1.1 Verifying the integrity of the download

The integrity of the installer image can be verified by comparing a computed hash value of the downloaded file against a hash computed by the pfSense project when the files were originally created. The current hashes provided by the project use SHA-256.

The SHA-256 sum displayed on the download page is the best source, as it is not pulled from the same directory as the download images. A file containing the SHA-256 sum is also available on the mirrors with the same filename as the chosen installer image, but ending in `.sha256`.

Use the accompanying SHA-256 sum from the download site or `.sha256` file to verify that the download successfully completed and is an official release of pfSense software.

Warning: The SHA-256 sums are computed against the compressed versions of the downloaded files. Compare the hash *before* decompressing the file.

Hash calculation programs vary by operating system, some common examples include:

Windows

Use HashTab to compare the value against the provided hash. With HashTab installed, right click on the downloaded file to access the **File Hashes** tab containing the **SHA256** hash, among others.

Tip: If a SHA256 hash is not displayed, right click in the hash view and click **Settings**, then check the box for **SHA256** and click **OK**.

The generated SHA256 hash from HashTab can be compared with the contents of the provided checksum.

Note: It is also possible to use the Linux `sha256sum` or `md5sum` commands within Cygwin if the Cygwin command prompt is launched as an Admin user.

Mac OSX

Use the `shasum` or `md5` command line utilities to generate a hash of the downloaded file.

Example:

```
shasum -a 256 pfSense-CE-2.5.2-RELEASE-amd64.iso.gz
```

The generated SHA256 hash can be compared with the contents of the provided **.sha256** checksum.

Linux

Use the `sha256sum` or `md5sum` command line utilities to generate a hash of the downloaded file.

```
sha256sum pfSense-CE-2.5.2-RELEASE-amd64.iso.gz
```

The generated SHA256 hash can be compared with the contents of the provided **.sha256** checksum.

FreeBSD

Use the `sha256` or `md5` command line utilities to generate a hash of the downloaded file.

```
sha256 pfSense-CE-2.5.2-RELEASE-amd64.iso.gz
```

The generated SHA256 hash can be compared with the contents of the provided **.sha256** checksum.

7.2 Prepare Installation Media

The installation image downloaded in the previous section must first be transferred to the proper media. The files cannot be copied to media directly, but must be written using appropriate tools.

The primary difference between the USB memstick and ISO image is in how the images are written to an installation disk. Both types of images install pfSense® software to a target disk. Another difference is between the console types for the different USB memstick images. After installation, they each retain their appropriate console settings.

Note: If the target hardware does not have an optical drive and cannot boot from USB, install the software to the target disk using a different set of hardware. See *Alternate Installation Techniques* for more information.

7.2.1 Decompress the Installation Media

The installation disk image is compressed when downloaded to save bandwidth and storage. Decompress the file before writing this image to an installation disk.

The `.gz` extension on the file indicates that the file is compressed with `gzip`. The image can be decompressed on Windows using 7-Zip, or on BSD/Linux/Mac with the `gunzip` or `gzip -d` commands.

7.2.2 Writing the Install Media

Creating an installation disk requires a different procedure depending on the type of media. Follow the instructions in the appropriate section for the chosen media type.

Prepare a USB Memstick

Warning: Be extremely careful when writing pfSense® disk images! If the client PC contains other hard drives it is possible to select the wrong drive and overwrite a portion of that drive with the installer disk. This renders the disk completely unreadable except to certain disk recovery programs, if at all.

Connect the USB Memstick to the Workstation

Start by connecting the USB memstick to the workstation containing the installation media image.

Locate the device name that the client PC designates for the drive. The device varies by platform, here are a few examples:

- Linux: `/dev/sdX` where `X` is a lowercase letter. Look for messages about the drive attaching in the system log files or by running `dmesg`.
- FreeBSD: `/dev/daX` where `X` is a decimal digit. Look for messages about the drive attaching in the system log files or by running `dmesg`.
- Windows: the drive will be named after a single uppercase letter, e.g. `D`. Use Explorer or examine the system control panel and look at the available disks for one matching the drive.
- On Mac OS X: `/dev/diskX` where `X` is a decimal digit. Run `diskutil list` from a command prompt or use the GUI tool **Disk Utility**.

  Note: On Mac OS X, if the disk is named `diskX` then the device to pass to the writing utility is actually `rdiskX` which is much faster for these types of low-level operations.

Note: Also make sure the device name refers device itself rather than a partition on the device. For example, `/dev/sdb1` on Linux is the first partition on the disk, so it would be writing to a partition on the device and the drive may not end up being bootable. In that case use `/dev/sdb` instead so the disk image utility writes to the entire disk.

Cleaning the USB Memstick

This step is optional unless the image fails to write to the USB memstick.

The target drive may already contain partitions which can prevent it from being written properly by disk image tools. To get a fresh start, wipe all of the partitions from the disk. This can be done a few different ways in Windows or in UNIX.

Windows

The **Disk Management** interface in Windows is one way to delete the partitions from a disk but often it has the operation disabled. The simplest and most reliable method is to use `diskpart`.

- Start a command prompt (`cmd.exe`) as Administrator
- Run `diskpart`
- Enter `list disk` to show the disks connected to the client PC
- Locate the target USB memstick in the list and note its disk number
- Enter `select disk n` where `n` is the disk number of the target USB memstick from the list in the previous command output
- Enter `clean` to remove the partitions from the disk
- Enter `exit` to stop `diskpart` and return to a command prompt
- Enter `exit` again to close the command prompt window

Linux, FreeBSD, Mac OS X

The `dd` command is the easiest way to erase the partition table from the USB memstick on UNIX and UNIX-like operating systems such as Linux, FreeBSD, and OS X.

```
$ sudo dd if=/dev/zero of=memstick_disk_path bs=1M count=1
```

Replace `memstick_disk_path` with the path to the memstick disk device, e.g. `/dev/sdb`, `/dev/da1`, or `/dev/rdisk3`.

Write the Image

Now it is time to write the image to the USB memstick. The exact procedure varies by operating system.

Note: The following instructions assume the installation media image file has been decompressed by an appropriate utility first. For details, see *Decompress the Installation Media*.

Warning: The operations in this section will completely overwrite any existing content on the USB Memstick! Check the USB memstick first for any files to save or backup.

Linux, FreeBSD, Mac OS X

On Linux, FreeBSD and Mac OSX, write the image to the drive using the `dd` command. It takes this general form:

```
dd if=image_file_name of=usb_disk_device_name
```

Writing to the disk in this way generally requires elevated privileges, so the user writing the image will most likely need to use `sudo` to run the command.

Example `dd` disk writing commands:

- Linux:

```
sudo dd if=pfSense-CE-memstick-ADI-2.4.4-RELEASE-p3-amd64.img of=/dev/sdb bs=4M
```

- FreeBSD:

```
sudo dd if=pfSense-CE-memstick-ADI-2.4.4-RELEASE-p3-amd64.img of=/dev/da1 bs=4m
```

- Mac OSX:

```
sudo dd if=pfSense-CE-memstick-ADI-2.4.4-RELEASE-p3-amd64.img of=/dev/rdisk3 bs=4m
```

The `bs=X` parameter is optional and tells `dd` to perform reads and writes on 4 MB blocks of data at a time. The default block size used by `dd` is 512 bytes. Specifying a larger block size can significantly increase the writing speed.

Windows

In order to write an image to a drive from a Windows workstation, use a GUI tool such as Win32 Disk Imager or Rufus. The same Linux `dd` command listed above can also be used from within Cygwin if the Cygwin command prompt is launched as an Administrator-level user.

Win32 Disk Imager

- Download and install Win32 Disk Imager
- Start Win32 Disk Imager as **Administrator**
- Click the folder icon
- Navigate to the location of the decompressed installation media image
- Select the image
- Choose the target USB memstick drive from the **Device** drop-down
- Click **Write**
- Wait for the image to finish writing

Rufus

- Download and install Rufus
- Start Rufus as **Administrator**
- Choose the target USB memstick drive from the **Device** drop-down
- Select *DD Image* from the drop-down next to **Create bootable disk using**
- Click the CD-ROM icon next to **Create bootable disk using**
- Navigate to the location of the decompressed installation media image
- Select the image
- Click **Start**
- Wait for the image to finish writing

Troubleshooting

If writing the disk fails, especially on Windows, clean the USB memstick as suggested in *Cleaning the USB Memstick* then try again. If it still fails, try a different USB memstick.

Prepare a DVD

To use an ISO image file containing pfSense® software with an optical disk drive, the ISO image must be burned to a DVD disc by appropriate writing software.

Since the ISO image is a full-disc image, it must be burned appropriately for image files **not** as a data DVD containing the single ISO file. Burning procedures vary by OS and available software.

Decompress the ISO Image

Before the image can be burned, it must be decompressed. The `.gz` extension on the file indicates that it is compressed with `gzip`. This can be decompressed on Windows using 7-Zip, or on BSD/Linux/Mac with the `gunzip` or `gzip -d` commands.

Burning in Windows

Windows 7 and later include the ability to burn ISO images natively without extra software. On top of that, virtually every major DVD burning software package for Windows includes the ability to burn ISO images. Refer to the documentation for the DVD burning program. A Google search with the name of the burning software and `burn iso` also helps locate instructions.

Burning with Windows

To burn a disc image natively in Windows 7 or later:

- Open Windows Explorer and locate the decompressed ISO image file
- Right click the ISO image file
- Click **Burn disc image**
- Select the appropriate **Disc burner** drive from the drop-down list
- Insert a blank DVD disc
- Click **Burn**

Later versions such as Windows 10 also show a **Disc Image Tools** tab on the ribbon when an ISO image is selected in Windows Explorer. That tab has a **Burn** icon that also invokes the same disc burning interface.

Other Free Burning Software

Other free options for Windows users include ISO Recorder, CDBurnerXP, InfraRecorder and ImgBurn. Before downloading and installing any program, check its feature list to make sure it is capable of burning an ISO image.

Burning in Linux

Linux distributions such as Ubuntu typically include a GUI DVD burning application that can handle ISO images.

If a DVD burning application is integrated with the window manager, try one of the following:

- Right click on the decompressed ISO image file
- Choose **Open With**
- Choose **Disk image writer**

Or:

- Right click on the decompressed ISO image file
- Choose **Write disc to**

Other popular applications include K3B and Brasero Disc Burner.

If a GUI burning program is not available, it may be possible to burn from the command line.

First, determine the burning device's SCSI ID/LUN (Logical Unit Number) with the following command:

```
$ cdrecord --scanbus
scsibus6:
      6,0,0   600) 'TSSTcorp' 'CDDVDW SE-S084C ' 'TU00' Removable CD-ROM
```

Note the SCSI ID/LUN is `6,0,0` in this example.

Burn the image as in the following example, replacing `<max speed>` with the speed of the burner (e.g. `24`) and `<lun>` with the SCSI ID/LUN of the recorder:

```
$ sudo cdrecord --dev=<lun> --speed=<max speed> pfSense-CE-2.4.4-RELEASE-p3-amd64.iso
```

Burning in FreeBSD

FreeBSD can use the same `cdrecord` options as Linux above by installing `sysutils/cdrtools` from ports or packages, and can also use GUI applications such as K3B or Brasero Disc Burner if they are installed from ports.

See also:

For more information on creating DVDs in FreeBSD, see the DVD burning entry in the FreeBSD Handbook.

Verifying the Disc

After writing the disc, verify it was burned properly by viewing the files on the disc. More than 20 folders should be visible, including `bin`, `boot`, `cf`, `conf`, and more. If only one large ISO file is visible, the disc was not burned properly. Repeat the burning steps listed earlier and be sure to burn the ISO file as a DVD image and **not** as a data file.

7.3 Perform the Installation

This section describes the process of installing pfSense® software to a target drive, such as an SSD or HDD. In a nutshell, this involves booting from the installation memstick or CD/DVD disc and then completing the installer.

Note: If the installer encounters an error while trying to boot or install from the installation media, see *Troubleshooting Installation Issues*.

The following items are requirements to run the installer:

- *Download Installation Media*
- *Prepare Installation Media*
- *Connect to the Console*

See also:

Virtual environments may have additional requirements, see the following documents for examples:

- *Virtualizing pfSense with VMware vSphere / ESXi*
- *Virtualizing pfSense with Hyper-V*
- *Virtualizing with Proxmox® VE*

See also:

pfSense Hangouts on Youtube also covers a variety of relevant topics.

7.3.1 Booting the Install Media

For USB memstick installations, insert the USB memstick and then power on the target system. The BIOS may require the disk to be inserted before the hardware boots.

For CD/DVD installations, power on the hardware then place the CD into an optical drive.

pfSense will begin to boot and will launch the installer automatically.

Specifying Boot Order in BIOS

If the target system will not boot from the USB memstick or CD, the most likely reason is that the given device was not found early enough in the list of boot media in the BIOS. Many newer motherboards support a one time boot menu invoked by pressing a key during POST, commonly `Esc` or `F12`.

Failing that, change the boot order in the BIOS. First, power on the hardware and enter the BIOS setup. The boot order option is typically found under a **Boot** or **Boot Priority** heading, but it could be anywhere. If support for booting from a USB or optical drive is not enabled, or has a lower priority than booting from a hard drive containing another OS, the hardware will not boot from the installer media. Consult the motherboard manual for more detailed information on altering the boot order.

7.3.2 Installing to the Hard Drive

For USB memsticks with a serial console connection, the first prompt will ask for the terminal type to use for the installer. For PuTTY or GNU screen, `xterm` is the best type to use. The following terminal types can be used:

ansi Generic terminal with color coding

vt100 Generic terminal without color, most basic/compatible option, select if no others work

xterm X terminal window. Compatible with most modern clients (e.g. PuTTY, screen)

cons25w FreeBSD console style terminal

For VGA consoles, `cons25w` is assumed by the installer.

Note: To accept all of the defaults and use a typical installation, press `Enter` at each prompt until the installer finishes.

Once the installer launches, navigating its screens is fairly intuitive, and works as follows:

- To select items, use the arrow keys to move the selection focus until the desired item is highlighted.
- For installer screens containing a list, use the `up` and `down` arrow keys to highlight entries in the list. Use the `left` and `right` arrow keys to highlight the actions at the bottom of the screen such as **Select** and **Cancel**.
- Pressing `Enter` selects an option and activates the action associated with that option.

Starting the Installer

The installer contents are the same for both console types. The following document walks through the installation process in its entirety.

Installation Walkthrough

When the installer starts the first screen it presents offers license terms to accept.

Read the terms carefully then press `Enter` to **Accept** the terms and proceed.

Rescue Options

First, the installer prompts to launch rescue options or start the **Install** process.

Use the arrow keys to select an option, then press `Enter`. The options on this screen are:

Install Continue installing pfSense software

Rescue Shell Starts a basic shell prompt where advanced users can perform tasks to prepare the system in ways not fully supported by the installer, or to perform diagnostic tests or repairs on the firewall.

Recover config.xml Attempts to recover a pfSense software configuration file from a target disk in the system. See *Recover config.xml From Existing Installation* for details.

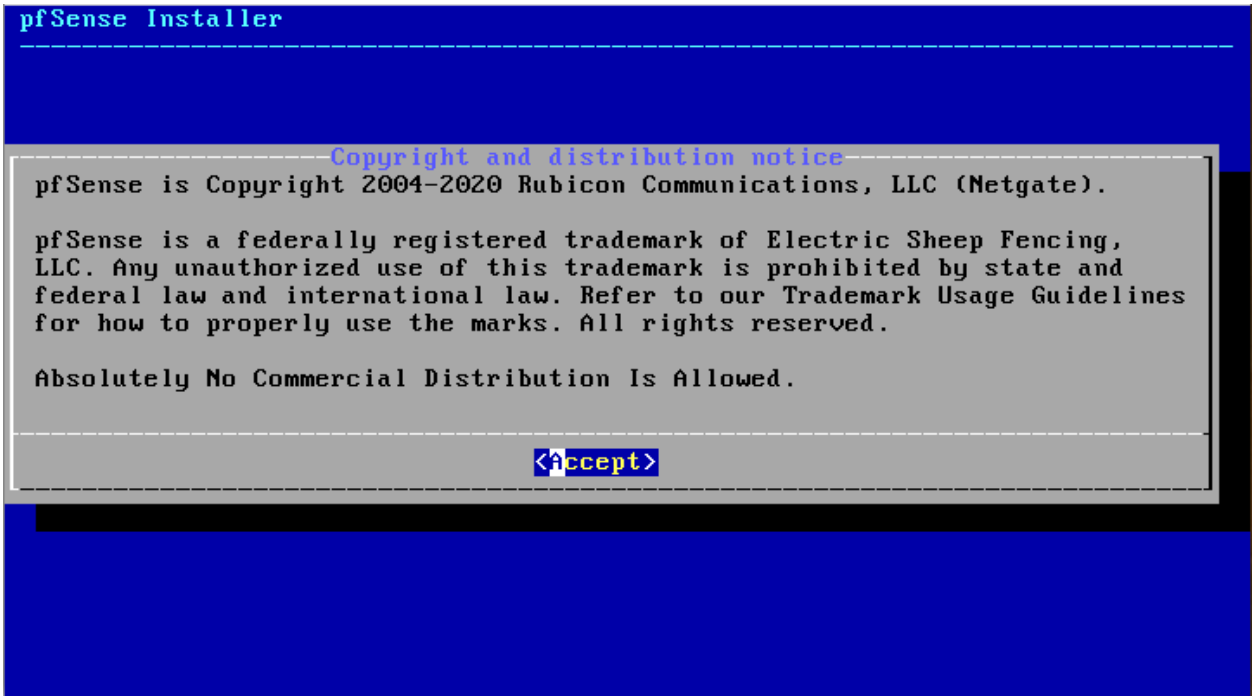


Fig. 1: Installer License

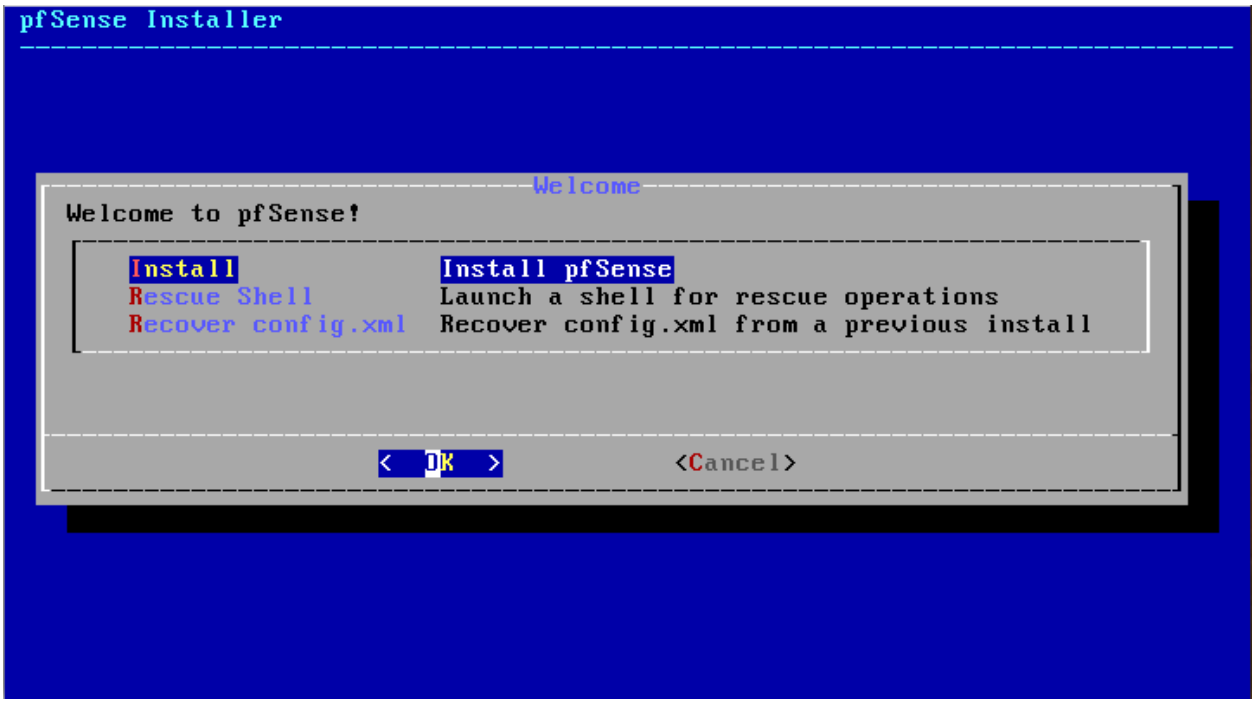


Fig. 2: Rescue Options

Keymap Selection

The **Keymap Selection** screen selects the keyboard layout used by the installer.

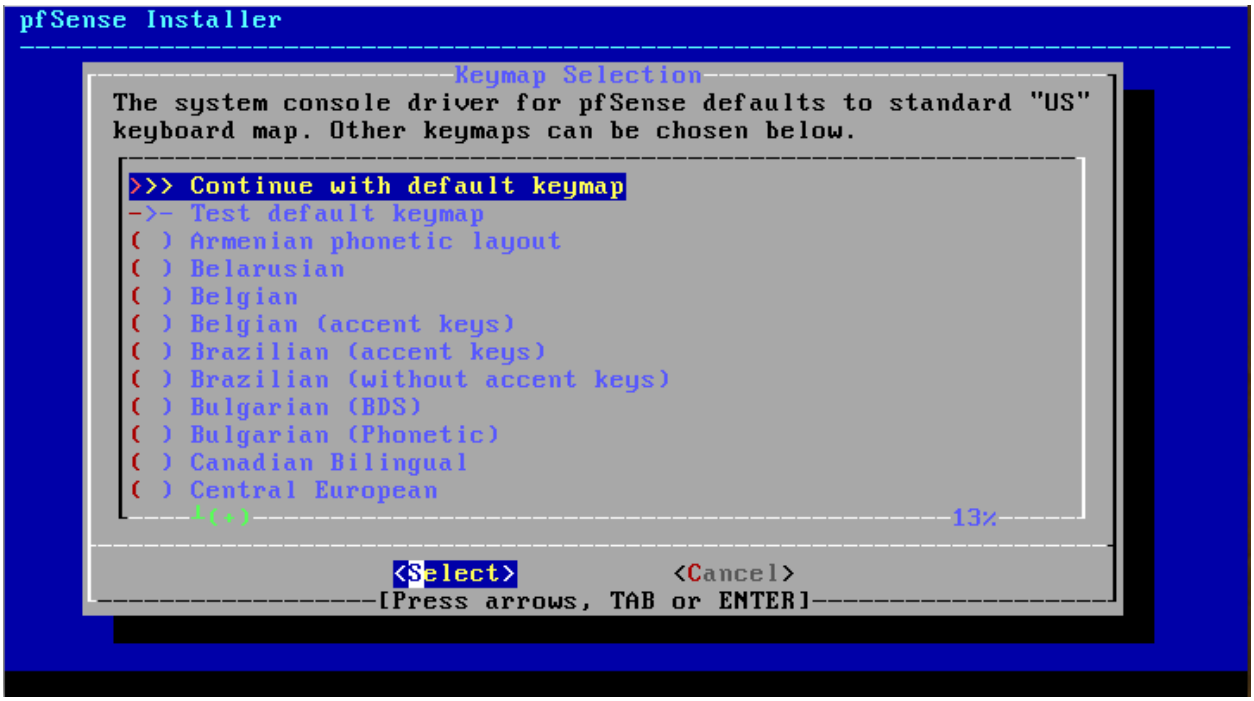


Fig. 3: Keymap Selection

For the majority of users with a standard PC keyboard, press `Enter` to select **Continue with default Keymap**. If the keyboard used for the console has a different layout, such as from countries other than the US, find it in the list and select it instead. After making a selection, return to the top of the list and either choose **Test default keymap** or **Continue**.

Note: This selection is only for the installer, the value is not retained post-install.

Partition / Filesystem Selection

The **Partitioning** step selects the filesystem for the firewall's target disk.

The **ZFS** filesystem type is more reliable and has more features than UFS, however ZFS can be memory hungry. Either filesystem will work on hardware with several GB of RAM, but if RAM usage is critical to other tasks that will run on this firewall, UFS is a more conservative choice. For hardware that requires UEFI, use ZFS.

The optionson this screen work as follows:

Auto (UFS) BIOS Automatically creates partitions and formats the disk with UFS and a traditional/legacy BIOS style boot environment.

Auto (UFS) UEFI Automatically creates partitions and formats the disk with UFS and a UEFI boot environment.

Note: Though systems which support UEFI natively should use this option in most cases, there

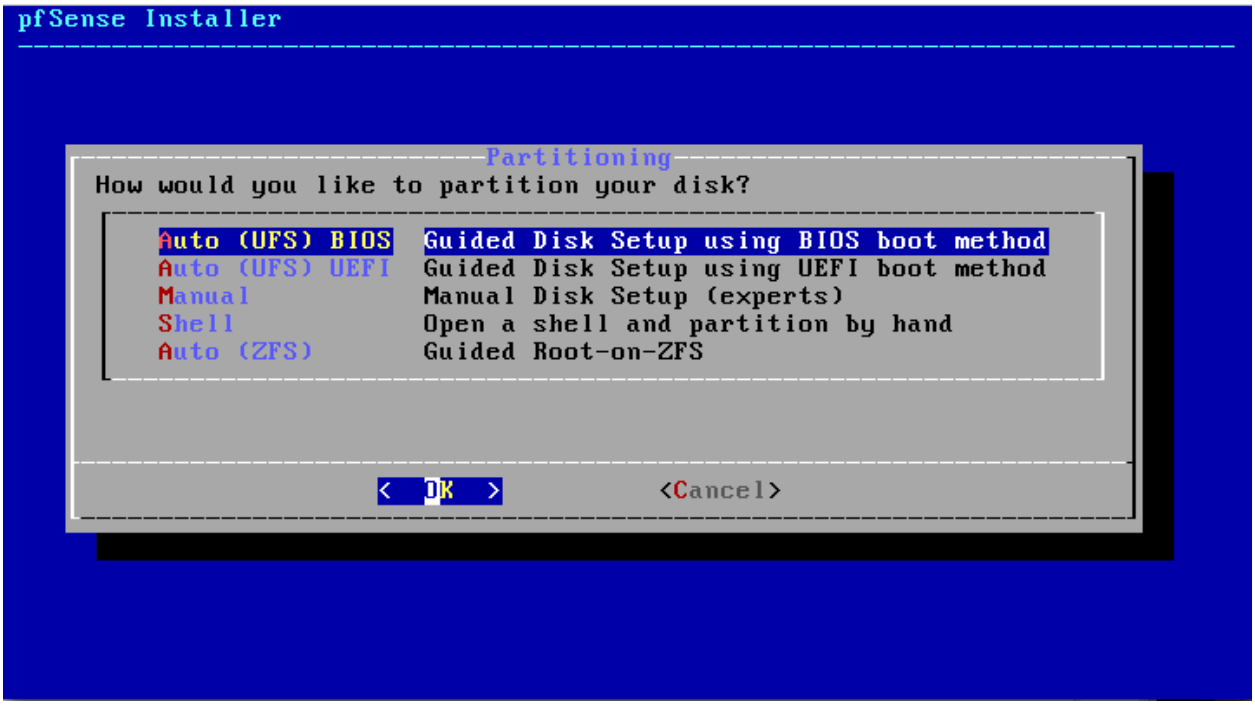


Fig. 4: Partitioning

are occasional incompatibilities between FreeBSD and UEFI implementations. If the system fails to boot, configure the system for BIOS/legacy booting and choose that installation option instead.

Manual Manually create partitions and filesystems.

Shell Open a shell prompt to configure disks, partitions, and filesystems by hand.

Auto (ZFS) Launches the ZFS configuration section of the installer. See *ZFS* for details.

Note: If installer cannot find any drives, or if it shows incorrect drives, it is possible that the desired drive is attached to an unsupported controller or a controller set for an unsupported mode in the BIOS. See *Troubleshooting Installation Issues* for help.

The process varies slightly depending on the selected filesystem type, so follow the section below that matches the filesystem type to be used by this firewall.

UFS

This section describes items specific to the UFS choices for partitioning.

Select **Auto (UFS) BIOS** or **Auto (UFS) UEFI** from the list (*Partitioning*) depending on the needs of the target system.

Single Disk

If there is only a single disk, the installation will perform the remaining steps automatically, there is nothing more to do, so proceed to *Continue with the Install*.

Multiple Disks

If the system has multiple eligible target disks, the installer will prompt to choose the target and other options.

Select Disk

Select the target disk where the installer will write out the pfSense software, e.g. `ada0`. The installer will show each supported hard drive attached to the firewall, along with any supported RAID or gmirror volumes.

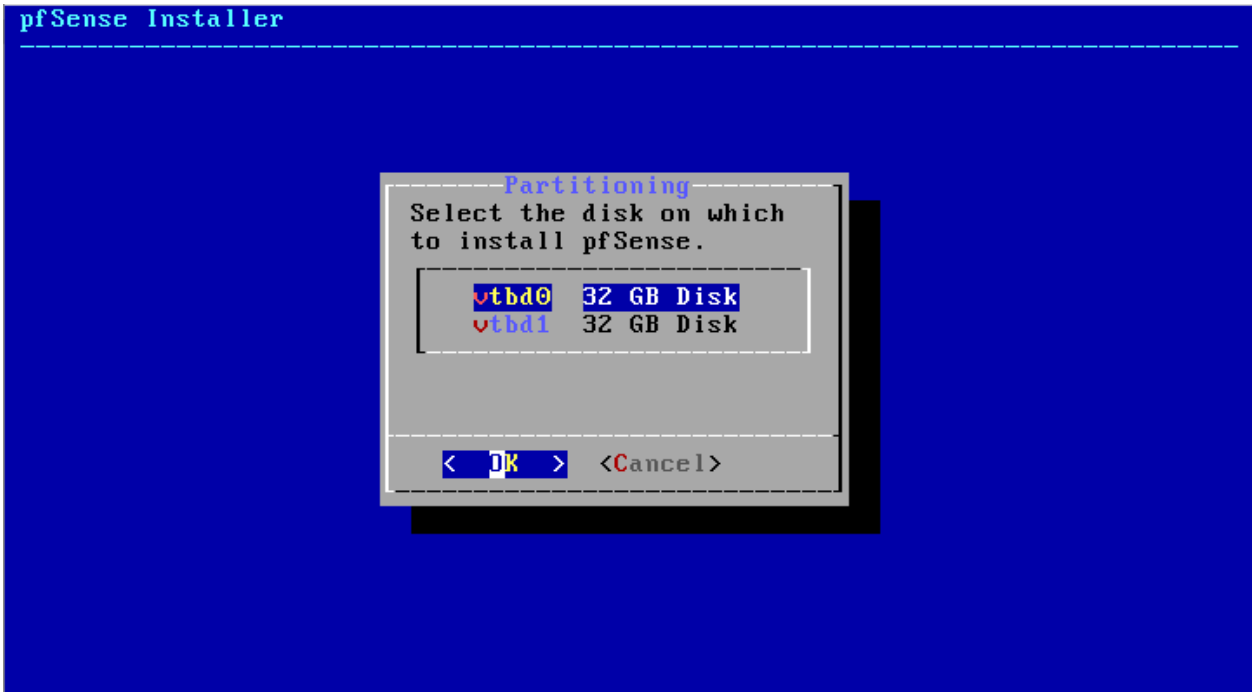


Fig. 5: Select Disk

Select **Entire Disk** when prompted. pfSense software does not support sharing a disk with another operating system.

Partition Scheme

Select the partition scheme to use for the disk:

GPT The GUID partition table layout. Used by most current x86 systems. May not function on older hardware/BIOS versions. **Try this method first**.

BSD BSD Labels without an MBR, which used to be known as "dangerously dedicated mode". This method should work on most hardware that cannot use GPT. This was the method used by older versions of pfSense software.

MBR Select this only if GPT and BSD do not work on a specific piece of hardware.



Fig. 6: Use the Entire Disk

Others The other choices are not relevant to hardware that is capable of running pfSense software.

Partition Editor

Select **Finish** to accept the automatic partition layout chosen by the installer, then select **Commit** to write the partition layout to the target disk.

Note: The partition sizes and other values can be customized here, but this is rarely necessary or appropriate. For nearly all installations, the default sizes are correct and optimal.

Finish

Proceed to *Continue with the Install*.

ZFS

This section describes items specific to ZFS partitioning.

Select **Auto (ZFS)** from the list (*Partitioning*) and the installer will present the ZFS configuration screen.



Fig. 7: Partition Scheme

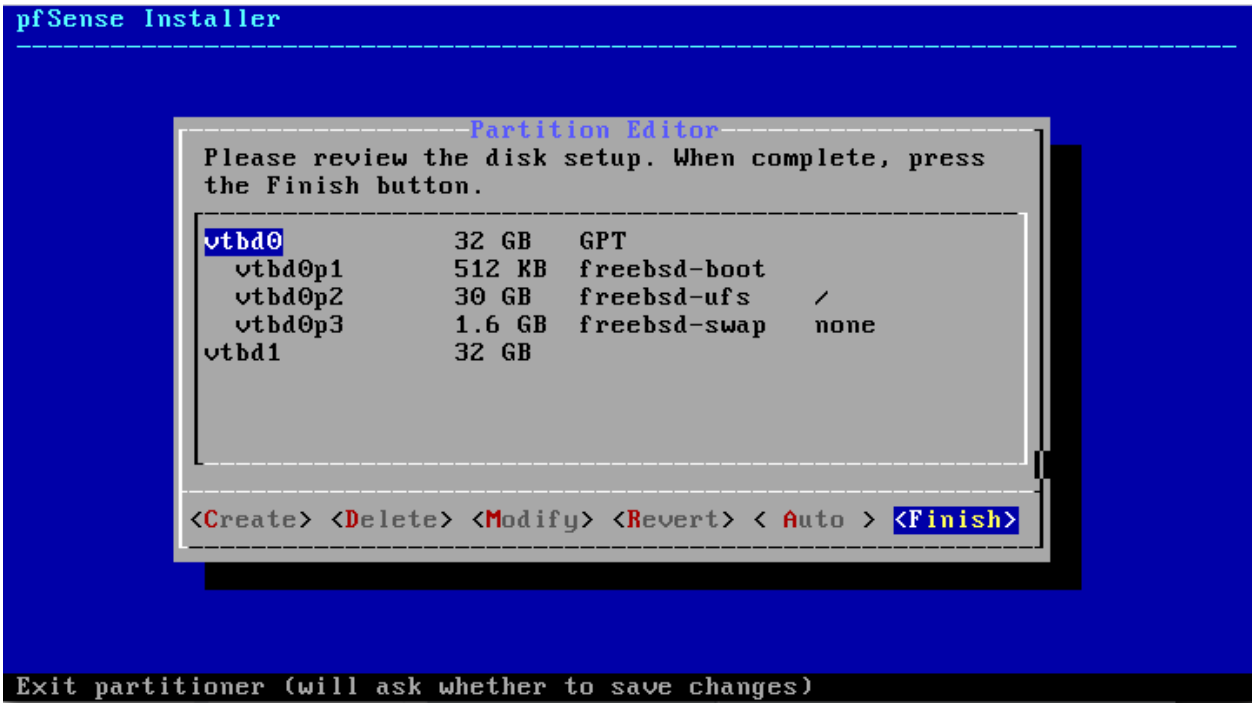


Fig. 8: Partition Editor

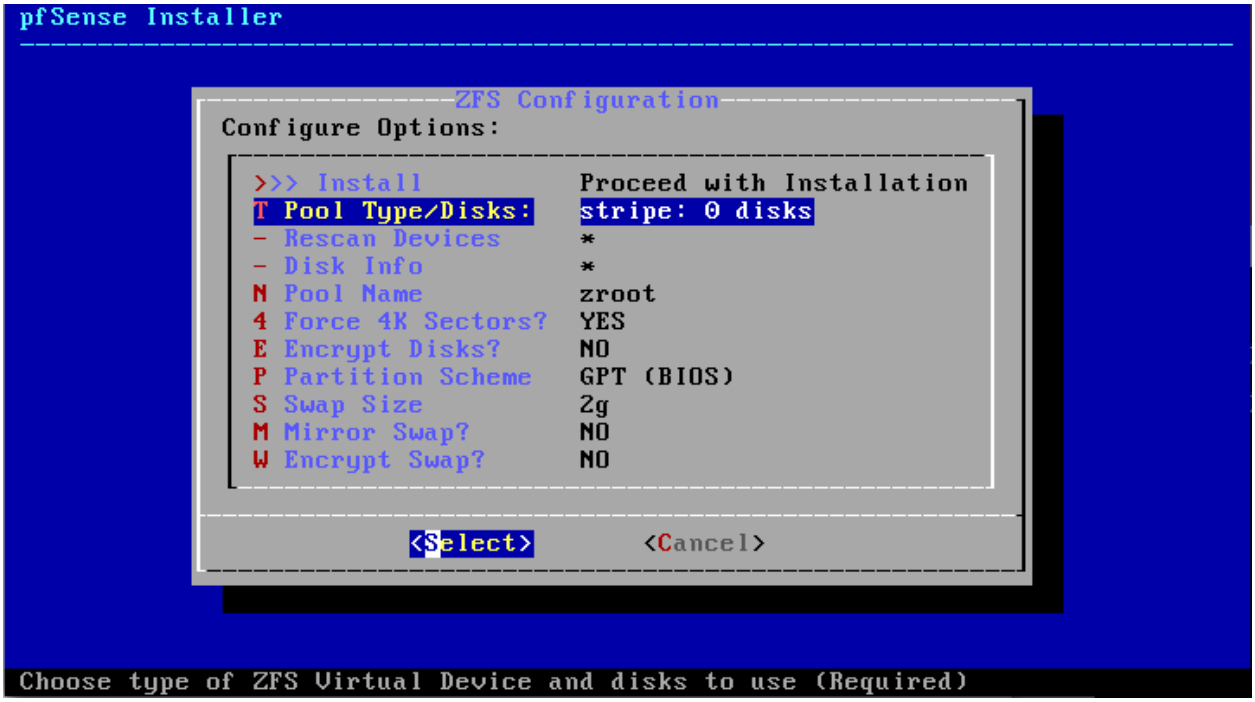


Fig. 9: ZFS Configuration

Pool Type / Disks

Select **Pool Type / Disks** and the installer will prompt for the **Virtual Device Type**. ZFS supports multiple disks in various ways for redundancy and/or extra capacity. Though using multiple disks with ZFS is software RAID, it is quite reliable and better than using a single disk.

The available types are:

stripe A single disk, or multiple disks added together to make one larger disk (RAID 0).

> **Note:** For firewalls with a single target disk, this is the correct choice.

mirror Two or more disks that all contain the same content for redundancy. Can keep operating even if one disk dies. (RAID 1)

raid10 RAID 1+0, n x 2-way mirrors. A combination of stripes and mirrors, which gives redundancy and extra capacity. Can lose one disk from any pair at any time.

raidzX Single, Double, or Triple redundant RAID. Uses 1, 2, or 3 parity disks with a pool to give extra capacity and redundancy, so either one, two, or three disks can fail before a pool is compromised. Though similar to RAID 5 and 6, the RAIDZ design has significant differences.

Select a type and press `Enter`

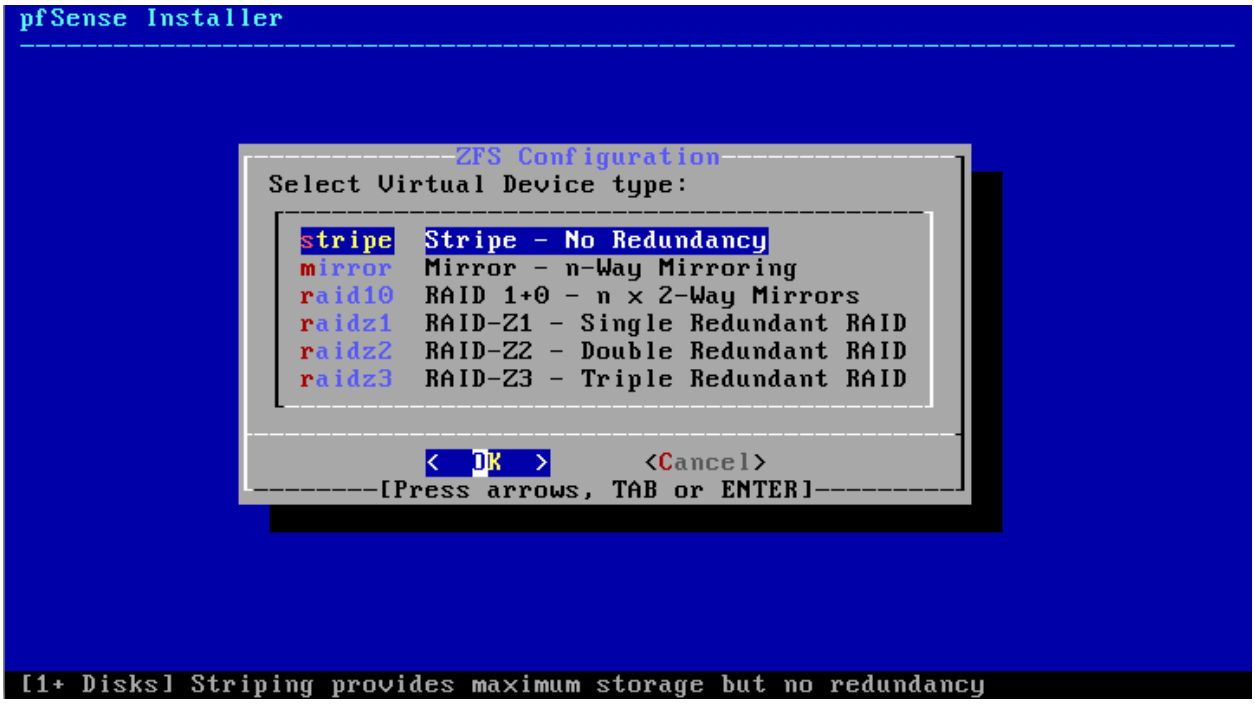


Fig. 10: Virtual Device Type

Select Disks

Next, the installer prompts for which disks it will include in the selected **Virtual Device Type**.

Use the `up` and `down` arrow keys to highlight a disk and `Space` to select disks. For mirrors or RAID types, select enough disks to fulfill the requirements for the chosen type.

Warning: Select a disk even if there is only one in the list!

Select OK with the `left` and `right` arrow keys.

When complete, the installer will return to the main ZFS configuration screen.

Partition Scheme

Choose an alternate **Partition Scheme** only if the default, GPT (BIOS) will not work. The possible choices include:

GPT (BIOS) The GUID partition table layout and BIOS booting. Used by most modern x86 systems.

Note: Try this method first as it is the most widely compatible choice.

GPT (UEFI) GPT with UEFI boot loader.

GPT (BIOS+UEFI) GPT with both BIOS and UEFI booting.

MBR (BIOS) Legacy MBR style partitions with BIOS booting.

GPT + Active (BIOS) GPT with the boot slice set active, with BIOS booting.



Fig. 11: Select Disks

GPT + Lenovo Fix (BIOS) GPT with a Lenovo-specific boot fix.

Other Options

The other options on the main ZFS configuration screen can be left at their default values. Certain scenarios may call for using them, but they are otherwise options.

Pool Name The name of the ZFS pool created by the installer. Unless there is an existing disk with the default name, `zroot`, leave this at the default.

Encrypt Disks Enable encryption of the filesystem contents.

> **Warning:** Encrypting disks will prompt for the encryption passphrase at each boot, which means each boot must be attended at the console.

Swap Size The amount of disk space dedicated to swap space (virtual memory). Typically the optimal size is 2x the available RAM in the firewall, but with smaller disks that may be too large.

Mirror Swap When using a mirrored Virtual Device Type, this also mirrors the swap space contents between disks. The default is to consider the swap space on each disk separately. In most cases the contents of swap are not important enough to warrant mirroring and the degraded performance it would impose.

Encrypt Swap Encrypts the contents of the swap partitions, in addition to data. This is more secure, especially if the firewall has particularly sensitive data in memory, but degrades swap performance.

Finish

To complete the installation:

- Move the selection back to **Install**
- Highlight **Select** for the action at the bottom of the screen
- Press `Enter` to continue.
- Select **Yes** to confirm the target disk selection, and to acknowledge that the contents of the target disk(s) will be destroyed.

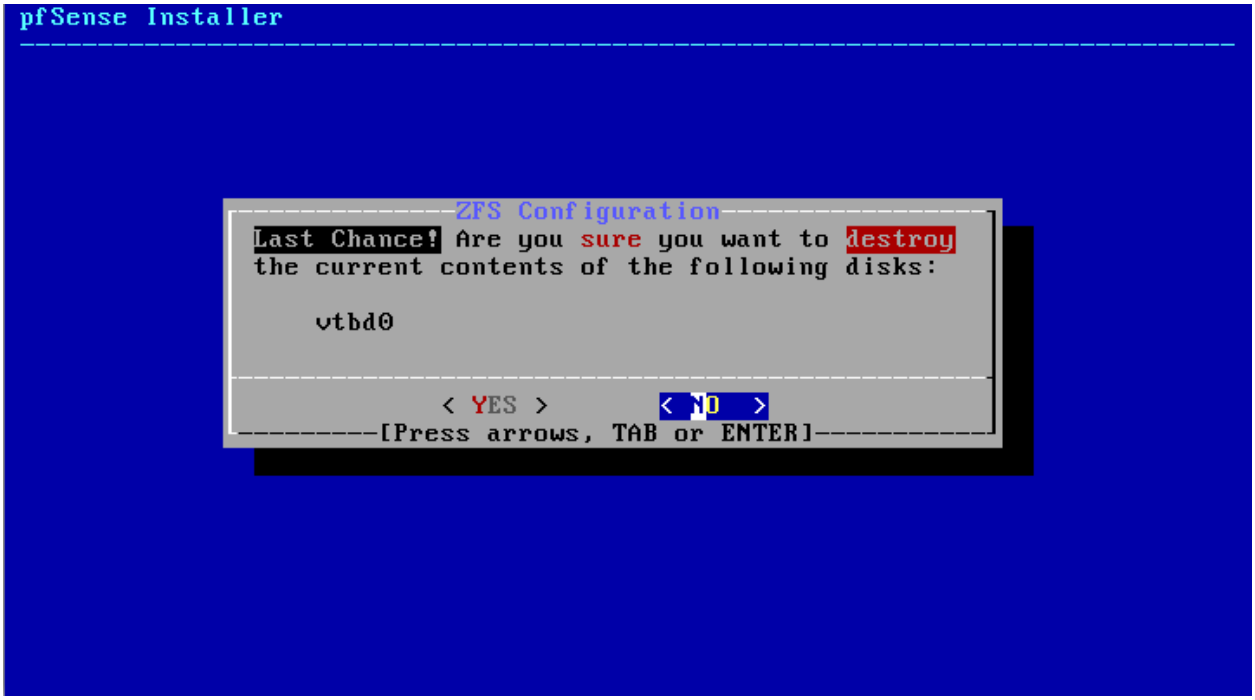


Fig. 12: ZFS Confirmation

Proceed to *Continue with the Install*.

Continue with the Install

Sit back, wait, and have a few sips of coffee while the installation process formats the drive(s) and copies pfSense files to the target disk(s).

Select **No** when prompted to make final modifications.

Select **Reboot** to restart the firewall

Remove the installation media from the firewall during the reboot, when the hardware is starting back up but before it boots from the disk.

Congratulations, the installation is complete!

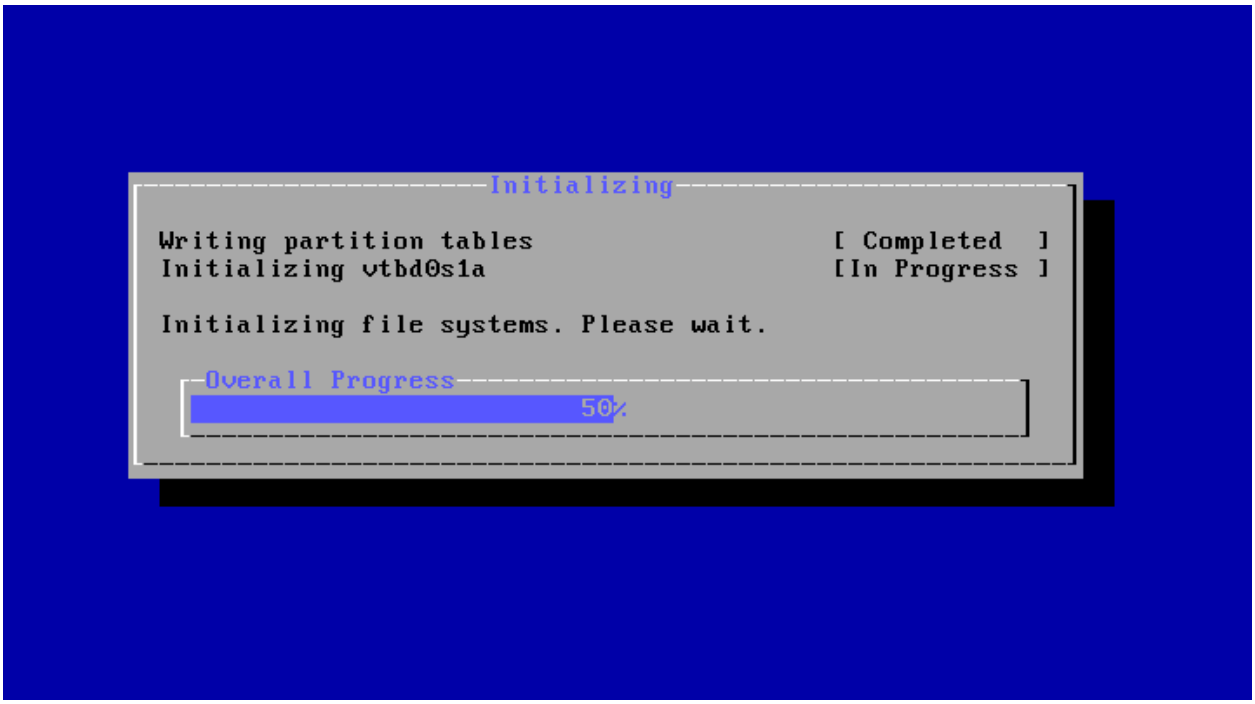


Fig. 13: Partitioning and Formatting

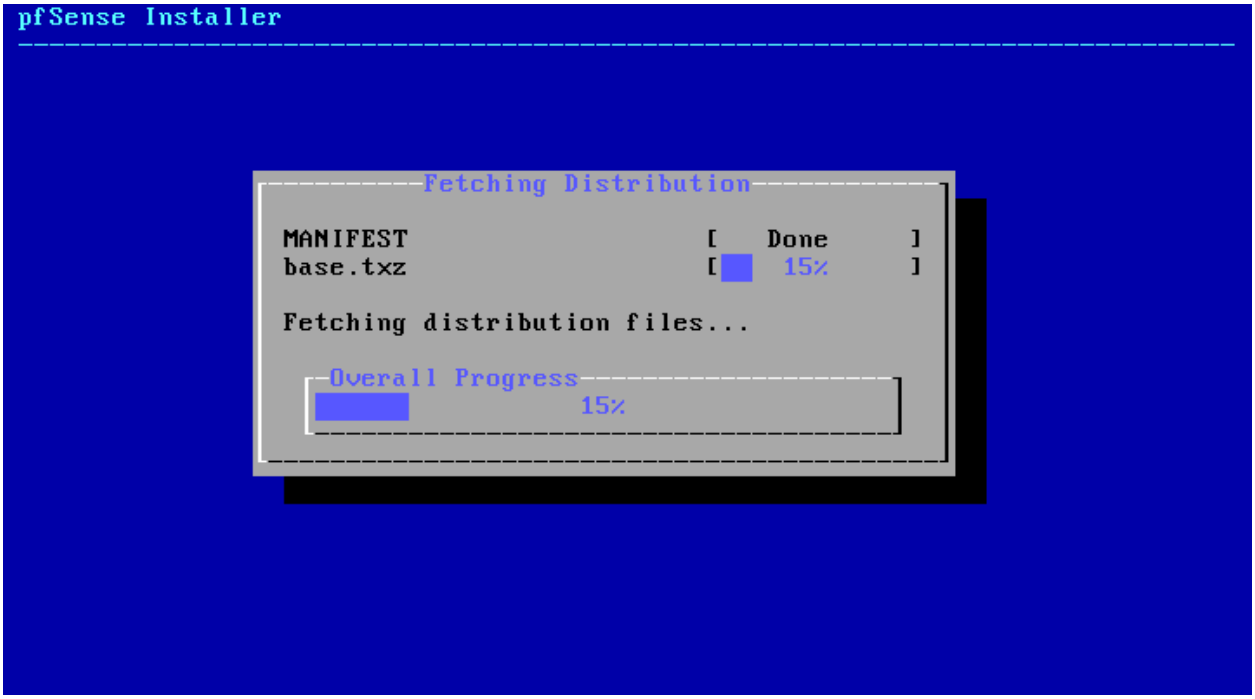


Fig. 14: Reading install media and copying to target drive

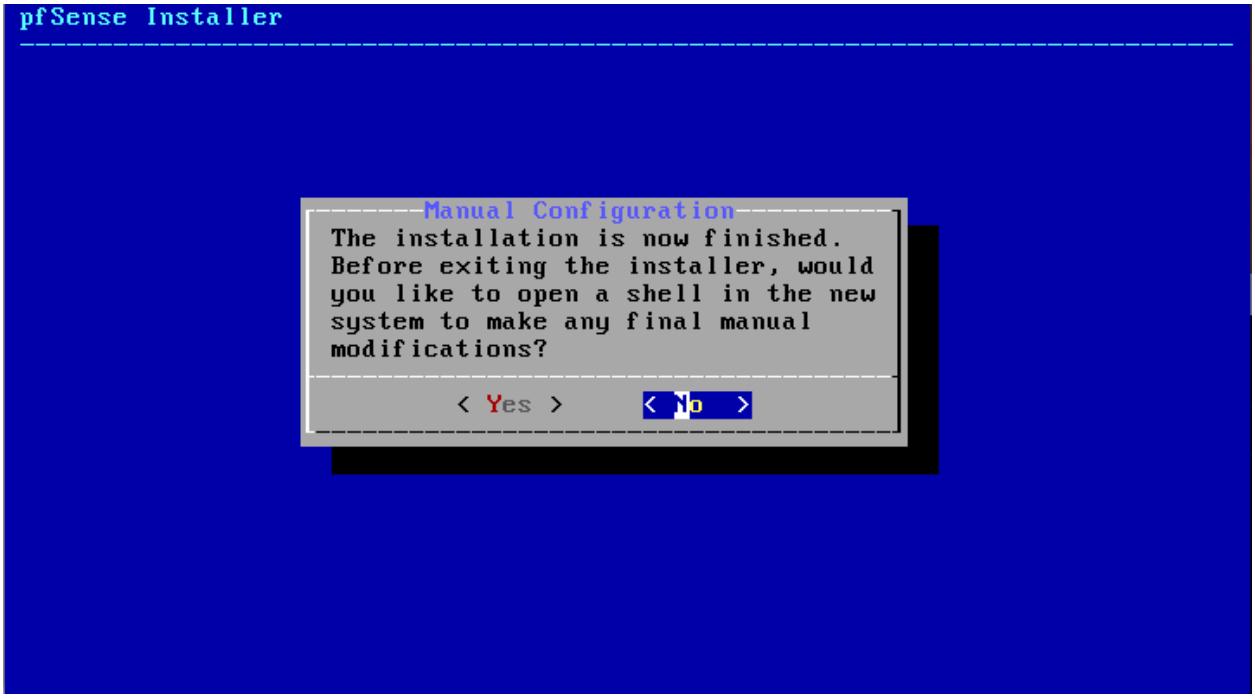


Fig. 15: Prompt for final modifications

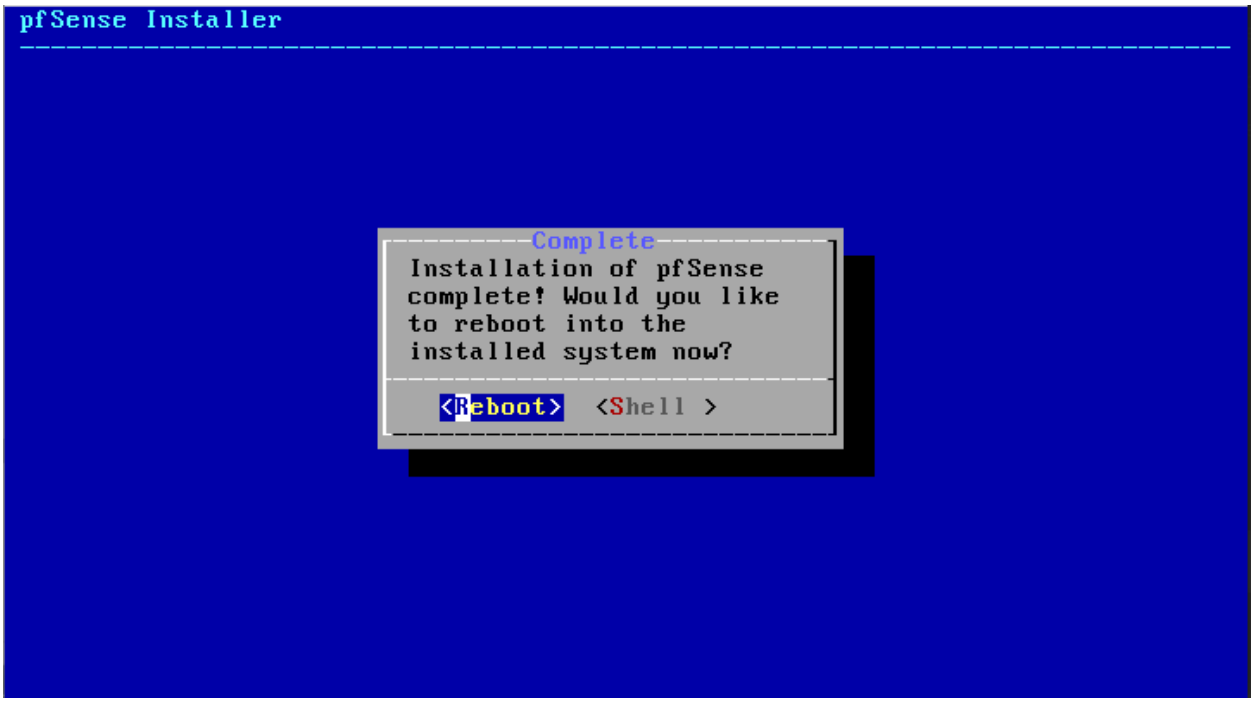


Fig. 16: Prompt to reboot

pfSense Default Configuration

After installation and interface assignment, pfSense has the following default configuration:

- WAN is configured as an IPv4 DHCP client.
- WAN is configured as an IPv6 DHCP client and will request a prefix delegation.
- LAN is configured with a static IPv4 address of **192.168.1.1/24**.
- LAN is configured to use a delegated IPv6 address/prefix obtained by WAN (Track IPv6) if one is available.
- All incoming connections to WAN are *blocked* by the firewall.
- All outgoing connections from LAN are *allowed* by the firewall.
- The firewall performs NAT on IPv4 traffic leaving WAN from the LAN subnet
- The firewall will act as an IPv4 *DHCP Server*
- The firewall will act as an IPv6 *DHCPv6 Server* if a prefix delegation was obtained on WAN, and also enables SLAAC
- The *DNS Resolver* is enabled so the firewall can accept and respond to DNS queries.
- SSH is disabled.
- WebGUI is running on port 443 using HTTPS.
- Default credentials are set to a username of `admin` with password `pfsense`.

7.4 Assign Interfaces

After the installer completes and the firewall reboots, the firewall software looks for network interfaces and attempts to assign interface mappings automatically.

The automatic interface assignment profiles used by the firewall are:

RCC-VE 4860/8860 WAN: igb1, LAN: igb0

RCC-VE 2220/2440 WAN: igb0, LAN: igb1

APU WAN: re1, LAN: re2

Other Devices For other devices the firewall looks for common interfaces and attempts to assign them appropriately, for example:

WAN: igb0, LAN: igb1

WAN: em0, LAN: em1

WAN: re1, LAN: re2

The factory firmware for devices from the Netgate Store includes additional default mappings appropriate to the hardware, which varies depending upon the hardware ordered with the device.

If the firewall cannot automatically determine the network interface layout, it will present a prompt for interface assignment as in Figure *Interface Assignment Screen*. This is where the network cards installed in the firewall are given their roles as WAN, LAN, and Optional interfaces (OPT1, OPT2 . . . OPTn).

The firewall displays a list of detected network interfaces and their MAC (Media Access Control) addresses, along with an indication of their link state if that is supported by the network card. The link state is denoted by `(up)` appearing after the MAC address if a link is detected on that interface.

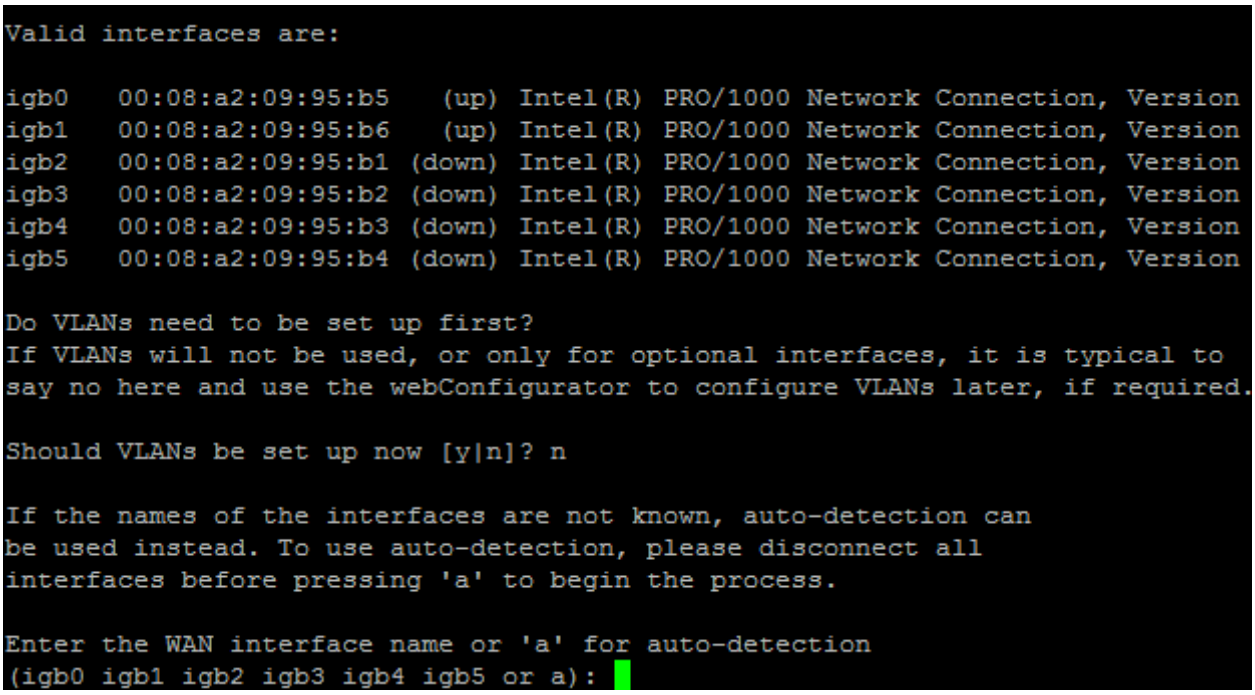


Fig. 17: Interface Assignment Screen

Note: The Media Access Control (MAC) address of a network card is a unique identifier assigned to each card, and no two network cards *should* have the same MAC address. If a duplicate MAC address is present on a network, either by chance or by intentional spoofing, all conflicting nodes will experience connectivity problems.

After printing the network interface list, the firewall prompts for VLAN configuration. If VLANs are desired, answer `y`, otherwise, type `n`, then press `Enter`.

See also:

For information about configuring VLANs, see *Virtual LANs (VLANs)*.

The firewall prompts to set the WAN interface first. As the firewall typically contains more than one network card, a dilemma may present itself: How to tell which network card is which? If the identity of each card is already known, enter the proper device names for each interface. If the difference between network cards is unknown, the easiest way to figure it out is to use the auto-detection feature.

For automatic interface assignment, follow this procedure:

- Unplug all network cables from the firewall
- Type `a` and press `Enter`
- Plug a network cable into the WAN interface of the firewall
- Wait a few moments for the firewall to detect the link up event
- Press `Enter`

If all went well, the firewall can determine which interface to use for the WAN.

Repeat the same process for the LAN and optional interfaces, if any are necessary. If the firewall prints a message stating “No link-up detected”, see *Manually Assigning Interfaces* for more information on sorting out network card identities.

Once the list of interfaces for the firewall is correct, press `Enter` at the prompt for additional interfaces. The firewall will ask **Do you want to proceed (y|n)?** If the network interface assignment list is correct, type `y` then press `Enter`. If the assignment is incorrect, type `n` and press `Enter` to repeat the assignment process.

Note: In addition to the normal routing/firewall mode with multiple interfaces, a firewall may also run in **Appliance Mode** where it has only a single interface (**WAN**). The firewall places the GUI anti-lockout rule on the WAN interface so a client may access the firewall web interface from that network. The usual routing and NAT functions are not active in this mode since there is no internal interface or network. This type of configuration is useful for VPN appliances, DHCP servers, and other stand-alone roles.

7.4.1 Manually Assigning Interfaces

If the auto-detection feature did not work, there is still hope of telling the difference between network cards prior to installation. One way is by MAC address, which the firewall prints next to the interface names on the assignment screen:

```
vmx0    00:0c:29:50:a4:04
vmx1    00:0c:29:50:ec:2f
```

The MAC address is sometimes printed on a sticker somewhere physically on the network card. For virtualized systems, the virtual machine configuration usually contains the MAC address for each network card. MAC addresses are assigned by manufacturer, and there are several online databases which offer reverse lookup functionality for MAC addresses in order to find the company which made the card: http://www.8086.net/tools/mac/, http://www.coffer.com/mac_find/, and http://aruljohn.com/mac.pl, among many others.

Network cards of different makes, models, or sometimes chipsets may be detected with different drivers. It may be possible to tell an Intel-based card using the `igb` driver apart from a Broadcom card using the `bge` driver by looking at the cards themselves and comparing the names printed upon the circuitry.

The probe order of network cards can be unpredictable, depending on how the hardware is designed. In a few cases, devices with a large number of ports may use different chipsets that probe in different ways, resulting in an unexpected order. Add-on and Multi-port NICs are generally probed in bus order, but that can vary from board to board. If the hardware has onboard NICs that are the same brand as an add-in NIC, be aware that some systems will list the onboard NIC first, and others will not. In cases when the probe order makes multiple NICs of the same type ambiguous, it may take trial and error to determine the port placements and driver name/number combinations.

After the network cards have been identified, type the name of each card at the interface assignment screen when prompted. In the above example, `vmx0` will be WAN and `vmx1` will be LAN. To assign them these roles, follow this procedure:

- Type `vmx0` and press `Enter` when prompted for the WAN address
- Type `vmx1` and press `Enter` when prompted for the LAN address
- Press `Enter` again to stop the assignment process, since this example does not contain any optional interfaces.
- Type `y` and press `Enter` to confirm the interface assignments

7.5 Alternate Installation Techniques

This section describes alternate methods of installation that may be easier for certain rare hardware requirements.

7.5.1 Installation with drive in a different machine

If it is difficult or impossible to boot from USB or from a DVD/CD drive to the target hardware, another computer may be utilized to install pfSense® software on the target hard drive. The drive may then be moved to the original machine.

After installation, allow the installation machine to restart and power it off once it returns to the BIOS screen. Remove the hard drive from the installation machine and place it into the target firewall. After boot, the firewall will prompt for interface assignment and then the rest of the configuration may be performed as usual.

Note: Current versions of pfSense software use techniques such as GPT id, UFS id, and ZFS metadata to mount disks, so even though the device may appear using a different disk driver on the actual target hardware, the OS will still be able to locate and mount the appropriate disk.

7.5.2 Installation in VMware with USB Redirection

USB redirection in VMware Player and Workstation can be used to install to a hard drive. Most any USB to SATA/IDE or similar adapter will work for this purpose. The following instructions are specific to VMware Workstation 12, but will also work on other recent versions.

- Plug the target drive into the SATA/IDE adapter or SD/CF writer
- Plug the adapter/writer into the client PC
- Open VMware Workstation on the client PC
- Create a *VM*, which should have USB enabled (It is enabled by default)
- Set the *VM* to connect the installer ISO image at boot in its virtual CD/DVD drive
- Start the virtual machine
- Press `Esc` during the *VM* BIOS screen to load the boot menu
- Find the icon for the USB adapter in the bottom of the VMware window
- Click the icon for the USB adapter
- Click **Connect (Disconnect from host)**
- Select **CD-ROM Drive** from the boot menu
- Continue through the installation the same as a normal, ensure that the correct drive is selected during the installation process
- Shut down the *VM*
- Remove the target disk from the client PC
- Attach the target disk to the intended firewall hardware

Older versions of VMware workstation can use automatic USB redirection to accomplish the same goal. Unplug the USB device, click inside the *VM* to give it focus, and then plug in the USB device. The *VM* should attach to the USB drive.

7.6 Upgrade Guide

pfSense® software can be reliably upgraded from an older release to a current release.

Netgate periodically release new versions that contain new features, updates, bug fixes, and various other changes. In most cases, updating an installation is easy. If the firewall is updating to a new release that is a only a point release (e.g 2.x.3 to 2.x.4), the update is typically minor and unlikely to cause problems.

Note: Only the most recent stable release of pfSense is officially supported, so updating is also important to ensure that any problems encountered may be resolved as needed.

The most common problems encountered during upgrades are hardware-specific regressions from one FreeBSD version to another, though those are rare. Updated releases fix more hardware than they break, but regressions are always possible. Larger jumps, such as from 2.3.x to 2.5.2-RELEASE must be handled with care, and ideally tested on identical hardware in a test environment prior to use in production.

Warning: Firewalls must be connected to the Internet to update.

7.6.1 Pre-Upgrade Tasks

Make a Backup . . . and a Backup Plan

Before making any modifications to a firewall, the best practice is to make a backup using the WebGUI:

- Navigate to **Diagnostics > Backup/Restore**
- Set the **Backup Area** to *All* in the **Backup Configuration** section of the page
- Click  **Download**
- Save this file somewhere safe

Keep multiple copies of the backup file in different secure locations. Consider using the free **Auto Config Backup** service (*Using the AutoConfigBackup Service*). Auto Config Backup can create a manual backup with a note identifying the change, which is encrypted and stored on Netgate servers.

Another good practice is to have install media handy for the new release, in case something goes awry and a reinstall is required. Should that happen, have the backup file on hand and refer to *Backup and Recovery*.

VM Snapshots

An easy fall-back plan for virtualized firewalls is to take a snapshot of the VM before performing an upgrade. This way, it can easily roll back to a known-good state if the VM encounters a problem.

Note: Before rolling back a VM due to problems, ensure the hardware compatibility of the VM is current. For example, on ESX 6.7, ensure the hardware compatibility is set to **ESXi 6.7 and later (VM version 14)** and update the VM Guest operating system to match the upgraded OS, such as *Other/FreeBSD 11 (64-bit)*

Pre-Upgrade Reboot

Reboot the firewall before applying an update. This step is optional, but a best practice.

If the hardware has a problem, such as a disk issue, then performing a reboot before the upgrade will allow that to be identified early. Otherwise, a hardware issue could be confused with an issue that occurred as a result of the upgrade process.

There is still a chance that the upgrade could draw out a hardware issue, such as a disk failing from the writes that happen in the upgrade process, but that is much less common to see in practice.

Packages

Warning: **Do not upgrade packages before upgrading pfSense® software**. Either remove all packages or leave the packages alone before running the update.

The safest practice is to remove **all** packages before upgrading to a new release. The upgrade process will handle packages automatically, but packages are frequently a source of problems. To ensure a smooth upgrade, note the installed packages, remove them, perform the upgrade, and then reinstall when the upgrade is complete.

7.6.2 Version-Specific Notes

This document covers specific concerns which must be taken into account by administrators when upgrading pfSense® software from an older version.

Review sections for each intermediate version between the version running on the firewall and the current release.

- *Upgrading from versions older than pfSense Plus 21.02.2 or pfSense CE 2.5.1*
- *Upgrading from versions older than pfSense 2.5.0*
- *Upgrading from versions older than pfSense 2.4.5-p1*
- *Upgrading from versions older than pfSense 2.4.4*
- *Upgrading from versions older than pfSense 2.4.0*
- *Older Version Upgrade Notes*

Upgrading from versions older than pfSense Plus 21.02.2 or pfSense CE 2.5.1

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

Upgrading from versions older than pfSense 2.5.0

- The built-in `relayd` load balancer has been deprecated and removed as it does not compile or run on pfSense 2.5.0. A copy of the load balancer configuration will be left in `/conf/deprecated_load_balancer.xml` for reference when converting to an alternate solution, such as HAProxy (*HAProxy package*).
- PHP was migrated from PHP 7.2 to PHP 7.4. A number of PHP errors were fixed along the way but certain combinations of configuration parameters may result in further errors. Note any problems on the Netgate Forum or the pfSense subreddit, and if possible, try to include relevant portions of `config.xml` with personal data removed.
- Due to the significant nature of the changes in this version of pfSense software, warnings and error messages, particularly from PHP and package updates, are likely to occur during the upgrade process. These errors are primarily seen on the console as the upgrade is applied, but may appear in a crash report once the upgrade completes. In nearly all cases these errors are a harmless side effect of the changes between FreeBSD 11.2 and 12.x and between PHP 7.2 and PHP 7.4.
- See the FreeBSD 12 Release Notes for information on deprecated hardware drivers that may impact firewalls upgrading to pfSense version 2.5.0. Some of these were renamed or folded into other drivers, others have been removed, and more are slated for removal in FreeBSD 13 in the future.
- OpenSSL was upgraded to 1.1.1a as a part of upgrading to FreeBSD 12.0, this will impact all packages which depend on OpenSSL, especially those not obtained from Netgate. Be aware that this will require obtaining new versions of such packages after the upgrade.

Upgrading from versions older than pfSense 2.4.5-p1

- Upgrading to pfSense software version 2.4.5-p1 requires `pfSense-upgrade` version 0.70 or later. Most installations will automatically pick up the new version and upgrade normally. If this does not happen automatically and the upgrade to version 2.4.5-p1 is not offered, use the following procedure:
  - Navigate to **System > Updates**
  - Set **Branch** to *Previous stable version*
  - Wait a few moments for the upgrade check to complete
  - Optional: Confirm that the latest version of *pfSense-upgrade* is present (version >= 0.70) using `pkg-static info -x pfSense-upgrade`.

    If the correct version is not present, wait a bit longer and check again as that package may be updating in the background.
  - Set **Branch** to *Latest stable version*
  - Wait a few moments for the upgrade check to complete

  At this point, the upgrade check should see 2.4.5-p1 and the upgrade can proceed.
- pfSense software version 2.4.5-p1 includes `pkg` version 1.13.x which introduces a new metadata version. Most installations will automatically pick up the new version and upgrade normally. In certain cases, especially coming from much older versions, the `pkg` utility may require a manual update before it can correctly process the new metadata.

  The `pkg` utility can be upgraded manually with the following command run from an ssh or console shell:

```
# pkg-static bootstrap -f
```

  See *Repository Metadata Version Errors* for more details.

Upgrading from versions older than pfSense 2.4.4

- Third party packages from **alternate repositories** are causing problems for users with the upgrade process and also with post-upgrade behavior. These packages have never been supported, and had to be manually added by users outside of the GUI.

  Due to the major changes required for FreeBSD 11.2 and PHP 7.2, third party packages from alternate repositories cannot be present during the upgrade. There is no way to predict if a third party package supports the new version or will cause the upgrade itself to fail.

  The upgrade process will automatically remove `pfSense-pkg-*` packages installed from alternate repositories. After the upgrade completes, the user can reinstall these packages. Packages from **alternate repositories** will not appear in the **Installed Packages** list in the GUI, and must be entirely managed in the command line.

  This change does not affect packages installed from the official pfSense package repository.

- *Using the AutoConfigBackup Service* is integrated into pfSense version 2.4.4 and free for all to use. It is no longer an add-on package. It is now located under **Services > Auto Config Backup**.
- PHP was migrated from PHP 5.6 to PHP 7.2. A number of PHP errors were fixed along the way but certain combinations of configuration parameters may result in further errors. Note any problems on the Netgate Forum or the pfSense subreddit, and if possible, try to include relevant portions of `config.xml` with personal data removed.
- Due to the significant nature of the changes in this version of pfSense software, warnings and error messages, particularly from PHP and package updates, are likely to occur during the upgrade process. These errors are primarily seen on the console as the upgrade is applied, but may appear in a crash report once the upgrade completes. In nearly all cases these errors are a harmless side effect of the changes between FreeBSD 11.1 and 11.2 and between PHP 5.6 and PHP 7.2.
- Gateway handling changes in 2.4.4 may result in different default gateway behavior than previous releases. Nearly all cases should behave properly, but be aware that it may be necessary to re-select the default gateway after upgrade.
- The FEC LAGG Protocol is deprecated and its options have been removed #8734
- The login protection daemon was changed from `sshlockout_pf` to `sshguard` and the behavior may be more sensitive in some cases to SSH and GUI login failures. For example, be aware of possible issues where probes from monitoring systems may end up triggering a block.
- Major changes to RADIUS for the base system and specifically Captive Portal could lead to behavior changes in certain cases. Read the release notes and associated bug reports for more details. Note any problems on the Netgate Forum or the pfSense subreddit.
- A crash report containing no data (empty) may appear after the upgrade completes. See #8915
- Intel Atom systems containing HD Graphics chipsets may experience console problems after the update. Affected systems will boot successfully, but fail to display console output after the boot menu. To fix the problem, add the following line to `/boot/loader.conf.local` to use the `syscons` console type:

```
kern.vty=sc
```

  - Alternately, try using i915 driver with the standard VT console using these lines in `/boot/loader.conf.local`:

```
i915kms_load="YES"
drm.i915.enable_unsupported=1
```

> **Warning:** This driver will consume extra bus resources and may cause resource hungry add-on hardware to fail, such as multi-port network adapters.

  - Systems with similar console problems not containing a graphics chip supported by the i915 driver may need to reinstall 2.4.4 to use a UEFI console.

- An ISP that supplies a bogus interface MTU via DHCP may cause interface problems with certain network interface types when **Advanced Configuration** options are present on DHCP interfaces, such as a DHCP WAN. The typical default case is handled automatically, but advanced options override the corrected default behavior. To fix the problem, apply the patch from #8507 or add `supersede interface-mtu 0` to the **Option modifiers** box in the WAN interface advanced DHCP options. If a custom `dhclient.conf` is in use, add `supersede interface-mtu 0` on a line inside the `interface` block. See #8507. The **Advanced Configuration** case has been corrected for the next release.

Upgrading from versions older than pfSense 2.4.0

- To use ZFS, a reinstall of the operating system is required. It is not possible to upgrade in-place from UFS to ZFS at this time.
- **Wireless** interfaces **must** be created on the **Wireless** tab under **Interfaces > Assignments** before they are available for assignment
- Some hardware devices may not boot 2.4.0 installation images, for example, due to UEFI compatibility changes. These are primarily BIOS issues and not issues with the installer images. Upgrading in place from 2.3.x typically allows affected hardware to run version 2.4.
- To upgrade Firewalls in place which are running pfSense software version 2.2.x or earlier, first upgrade the firewall to pfSense 2.3.4 and then perform an update to pfSense 2.4.x afterward. Alternately, reinstall 2.4.x directly and restore the configuration.

> **Warning:** When upgrading to 2.4.x from 2.2.x or earlier, remove all packages before attempting the update. Even when upgrading from 2.3.x this is the best practice to ensure a smooth upgrade process. Package settings are retained.

Older Version Upgrade Notes

Versions of pfSense software prior to 2.3 used a different upgrade method. For "full" installations, a `tgz` file was used by the firewall to copy in the new files. This method was problematic and is no longer used.

The best practice in these cases is to take a backup and reinstall with a current, supported version of pfSense software.

The following information is for upgrading from outdated and unsupported versions of pfSense software. They may still be of use to users attempting to upgrade from an older release to a current, supported, release.

When upgrading from a very old release, read every document below that covers versions between the older one being upgraded and the new version.

Upgrading from versions older than pfSense 2.3

See also:

For information about upgrading to current versions, see *Upgrade Guide*.

Warning: Uninstalling *all* packages is **required** when upgrading from old releases. Packages **must be removed** before the upgrade is performed. After the upgrade is complete, packages can be reinstalled. Package configuration is automatically retained.

See *2.3 New Features and Changes* for a larger list of changes.

- Due to the GUI overhaul, older themes have been removed. All previously chosen themes are reset on upgrade to the default "pfSense" 2.3 theme.
- **Status > RRD Graphs** moved to **Status > Monitoring** and has been revamped. The same data, and more, is still accessible but with a modern interface.
- **System > Firmware** is now **System > Update**
- **System > Packages** is now **System > Package Manager**

Limiters

- On pfSense® software versions 2.2 and 2.3, limiters cannot be used on firewall rules residing on interfaces where NAT applies. This limits their use to LAN-type interfaces only, and not WANs, in most circumstances. This has been fixed on pfSense 2.4. Bug #4326
- On pfSense software versions 2.2 and 2.3, limiters cannot be used where pfsync is enabled. This has been fixed on pfSense 2.4.3. Bug #4310

NanoBSD

Warning: NanoBSD has been deprecated as of pfSense 2.4.0-RELEASE. This section remains only for users on i386 hardware with NanoBSD who must upgrade to pfSense 2.3.5-p2.

In most cases, a normal installation may be used in place of NanoBSD. Activating the option to keep `/var` and `/tmp` in RAM can typically yield the same net benefits for older/slower CF and SD media. Firewalls with modern SSDs should have no concerns with writes.

1GB NanoBSD images have been removed as they were too small to properly function and upgrade. If a 1GB NanoBSD image is in use, it cannot be upgraded. It must be re-imaged on a larger card using the 4GB or 2GB image or converted to a full installation.

Package System

- Due to the package system overhaul, any custom package repository settings are removed so the firewall will pull package information directly from pfSense servers.
- We highly recommend uninstalling all packages before upgrading.

Removed features that are disabled on upgrade

- Groups with spaces are no longer permitted. They are not allowed at the OS level and were not functioning properly. On upgrade, such groups are renamed with an underscore ('_') in place of a space.
- The "Enable" checkbox for IPsec has been removed. If IPsec was disabled, all Phase 1 entries are disabled automatically on upgrade.
- The **Unity** plugin for IPsec has been disabled by default, where it was previously enabled by default. This is preferable for the vast majority of users, however those using mobile IPsec with IKEv1 may need to enable it under **VPN > IPsec**, **Advanced** tab.
- The `apinger` daemon for gateway monitoring has been replaced by `dpinger`. Due to the differences in settings between the two, many advanced gateway parameters are reset on upgrade.
- The PPTP *Server* has been removed, if the PPTP server was in use, seek alternate solutions such as IPsec or OpenVPN. **Do not continue to use PPTP**.
  - The PPTP server settings, firewall rules, and so on have all been removed
  - If the "Redirect" PPTP server type was in use, add manual NAT rules for TCP/1723 and GRE to point to the actual server.
- Layer 7 classification support has been removed and any configuration using L7 is automatically removed on upgrade.
- WEP support has been removed from Wireless interfaces, and if a wireless interface was using WEP, the interface is deactivated on upgrade.
- Single DES support has been removed from IPsec, if a Phase 1 or Phase 2 entry was using DES, it is deactivated on upgrade.
  - Note: 3DES support is still present. Only the older and insecure, **single** DES option was removed.
- The Live CD platform has been removed. The ISO is a bootable installer, as always, but it cannot run a live system.
  - For the very few people who were still using Live CD: If the hardware can boot from USB, install to a USB thumb drive and run from it instead. Use the options to keep `/var` and `/tmp` in RAM, and do not install packages, then net result should be similar but ultimately more functional.
- Some obsolete password hashes, such as nt-hash, are removed from users on upgrade. There was no remaining code on pfSense that utilized these hashes, so there should be no loss of functionality.
- Support for `fifolog` was removed, and will revert to clog format on upgrade.
- The `net.inet.ip.fastforwarding` tunable is no longer present, and is unset on upgrade.
- Some PHP modules, such as MySQL, were included by default on previous versions but are no longer a part of the base system on 2.3. They are available as packages that may be installed manually from the shell (e.g. `pkg install php56-mysql`)

New features that may require action

- The default system password hash has been changed to bcrypt. Current passwords will continue to work. Existing users need to reset their password to convert to the new, more secure, hash. #4120
- A new option was added to Captive Portal for FreeRADIUS-friendly stop/start RADIUS accounting updates that solves problems with user session time limits. If stop/start RADIUS accounting is being used with FreeRADIUS, the new option should be activated manually.

Upgrading from a 2.3 Snapshot

- If a firewall was upgraded to 2.3 before Jan 21, 2016, some files from 2.2.x or earlier packages may still be left behind that can prevent new packages from installing properly. Run the following command the clean up outdated symlinks that are not relevant for 2.3:

```
find / -type l -lname '/usr/pbi/*' -delete
```

Multi-WAN Weighted Load Balancing

There is a quirk in pf handling of weighted load balancing where Load balancing fails when one gateway has a weight of 1 and another gateway has a weight >1. Coming from 2.2.x, if this scenario applies, simply double the assigned weights. For example: WAN1 = `1`, WAN2 = `5` on 2.2.x should be WAN1 = `2`, WAN2 = `10` on 2.3.

Captive Portal

Due to the change in the web server from `lighttpd` to `nginx`, in some cases the portal HTML must be updated to include the zone parameter. On 2.3.1 and later the web server process attempts to handle this automatically, but it is best to include the HTML in the portal page directly, inside the form tag:

```
<input name="zone" type="hidden" value="$PORTAL_ZONE$">
```

Upgrading from versions older than pfSense 2.2

See also:

For information about upgrading to current versions, see *Upgrade Guide*.

Warning: Uninstalling **all** packages is **required** when upgrading from old releases. Packages **must be removed** before the upgrade is performed. After the upgrade is complete, packages can be reinstalled. Package configuration is automatically retained.

Limiters

- On pfSense® software versions 2.2 and 2.3, limiters cannot be used on firewall rules residing on interfaces where NAT applies. This limits their use to LAN-type interfaces only, and not WANs, in most circumstances. This has been fixed on pfSense 2.4. Bug #4326
- On pfSense software versions 2.2 and 2.3, limiters cannot be used where pfsync is enabled. This has been fixed on pfSense 2.4.3. Bug #4310

IPsec Changes

The IPsec daemon was changed from **racoon** to **strongSwan**. Existing configurations work the same as always, but if any unusual configurations are present, take care in testing after the upgrade. Changes in behavior because of this change may trigger bugs in remote endpoints that weren't previously an issue. Configurations that were always technically incorrect may exhibit problems now where they didn't previously. We have listed the circumstances we are aware of here, and will expand upon this list if anything new is found.

Problem in racoon with aggressive mode and NAT-D

Those using racoon (pfSense 2.1.x and earlier, among a variety of other similar products) on remote endpoints with aggressive mode may encounter a bug in racoon related to NAT-D and aggressive mode. Any site to site IPsec VPNs using aggressive mode with racoon as a remote endpoint should change to main mode to prevent this from being an issue. Main mode is always preferable for its stronger security.

glxsb Crypto Accelerator Warning

For those using the `glxsb` crypto accelerator in the ALIX and other devices with Geode CPUs, only AES 128 bit is supported by those cards. Any key length > 128 bit has never worked, and must not be configured. There appear to be circumstances where AES on "auto" with racoon preferred 128 bit where strongswan prefers the strongest-available and is choosing 256 bit, which glxsb breaks. Input validation in 2.2.1 prevents such invalid configurations when adding configurations or making changes, however existing configurations are not changed. If using glxsb and AES, ensure both phase 1 and phase 2 configurations all use AES 128 only and never auto.

Mobile client users, verify Local Network

For mobile IPsec clients, clients could pass traffic in certain circumstances without having specified the necessary matching local network in the mobile phase 2 configuration. The "Local Network" specified in mobile IPsec phase 2 must include all networks mobile clients need to reach. If mobile IPsec clients need to access the Internet via IPsec, the mobile phase 2 must specify `0.0.0.0/0` as the local network.

Stricter Phase 1 Identifier Validation

In 2.1.x and earlier versions, racoon could accept mismatched phase 1 identifiers where using *IP Address* as the identifier. This is most commonly a problem where one of the endpoints is behind NAT and phase 1 is using *My IP Address* and *Peer IP Address* for identifiers. On the side with the private IP WAN, *My IP Address* will be its private WAN IP address. On the opposite end, *Peer IP Address* will be the public IP address of the opposite side. Hence, these two values do not match, and should have resulted in a connection failure. racoon would fall back to checking the source IP address of the initiating host as an identifier, where it found the match. To resolve this issue, change the phase 1 identifiers so they actually match.

Phase 2 behavior change with incorrect network addresses

In 2.1.x and earlier versions a phase 2 configuration with an incorrect network address would still be presented by racoon with the corrected network address. e.g. if `192.168.1.1/24` is set in a phase 2, which should be `192.168.1.0/24`, racoon used it as `192.168.1.0/24`. In 2.2.x and newer versions, strongswan sends it exactly as configured. This may result in a phase 2 mismatch where configured with an incorrect network address.

Disk Driver Changes

The disk drivers in FreeBSD changed between the underlying OS versions and now the CAM-based ATA drivers and AHCI are used by default. As such, ATA disks are labeled as `/dev/adaX` rather than `/dev/adX`. The `ada` driver for ATA disks and GEOM keeps legacy aliases in place so that old disk references will still work post-upgrade. This does not always extend to virtualized disk drivers, however (see the Xen note below.). The upgrade process on pfSense 2.3 and 2.4 also attempts to automatically correct for this change.

A manual workaround is also possible. Running `/usr/local/sbin/ufslabels.sh` **before** the upgrade will convert `/etc/fstab` to UFS labels rather than disk device names bypassing any device name issues that could arise due to the switch.

There is a chance that the new driver stack will have issues with certain controller/disk combinations that were not present in prior releases. There may be BIOS changes or other workarounds to help. See *Boot Troubleshooting*.

The methods used to disable DMA and write caching have both changed on FreeBSD 10.x. For most, disabling these manually is no longer necessary.

If disabling DMA is necessary, the following may be used in `/boot/loader.conf.local`:

```
hint.ata.X.mode=PIO4
```

Change `X` to be the ATA controller ID, typically `0` or `1`.

If write caching must be disabled, the following may be used in `/boot/loader.conf.local`:

```
kern.cam.ada.write_cache=0
```

Xen Users

The FreeBSD base used by pfSense 2.2 and later includes PVHVM drivers for Xen in the kernel. This can cause Xen to automatically change the disk and network device names during an upgrade to pfSense 2.2 or later, which the Hypervisor should not do but does anyway.

The disk change can be worked around by running `/usr/local/sbin/ufslabels.sh` **before** the upgrade to convert `/etc/fstab` to UFS labels rather than disk device names.

The NIC device change issue has no workaround. Manual reassignment is required.

vmxnet3 (VMware/ESX) users

Users who manually installed *VMware Tools* to use `vmxnet3` network adapters may encounter an issue with interface name changes when upgrading to pfSense 2.2 or later, similar to those with Xen mentioned above. In pfSense 2.1.x the `vmxnet3` interfaces were named starting with `vmx3f` and on pfSense 2.2.x they are `vmx` using the built-in support. Manually reassigning the interfaces or correcting them in `config.xml` followed by a restore is required.

Old/Broken GEOM Mirrors

If a manual gmirror configuration was performed post-install and not using the pfSense installer gmirror option before install, there is a chance that the mirror will not function on pfSense 2.2 or later because the manual post-install method did not create a proper mirror setup. If an upgraded mirror does not boot or function on pfSense 2.2 or later, use the following entry to work around the integrity check that would otherwise fail.

Add the following line to `/boot/loader.conf.local`:

```
kern.geom.part.check_integrity=0
```

If the disks are configured in this way, we **strongly** recommend backing up the configuration and reinstalling, using one of the mirrored disk options in the pfSense installer.

CARP Changes

Due to the new CARP subsystem, the old method of having a virtual interface for CARP VIPs is no longer available. CARP VIPs work more like IP Alias style VIPs, existing directly on the main interface. For most, the changes made to accommodate this new system will be transparent, but there are some potential issues, such as:

- With no separate interface available, monitoring a CARP VIP status via SNMP is no longer possible.

FTP Proxy

The FTP proxy is not included in pfSense 2.2-RELEASE or later, due to changes in the kernel and state table handling that made it it more difficult to implement. Use of FTP is strongly discouraged as credentials are transmitted insecurely in plain text. #4210

See *FTP without a Proxy* for additional information and workarounds.

Another option is the recently added FTP Client Proxy package which leverages in FreeBSD to allow clients on local interfaces to reach remote FTP servers with active FTP.

LAGG LACP Behavior Change

LAGG using LACP in FreeBSD 10.0 and newer defaults to "strict mode", which means the lagg does not come up unless the attached switch is speaking LACP. This will cause a LAGG to not function after upgrade if the switch is not using active mode LACP.

To retain the lagg behavior in pfSense 2.1.5 and earlier versions, add a new system tunable under **System > Advanced**, **System Tunables** tab for the following:

```
net.link.lagg.0.lacp.lacp_strict_mode
```

With value set to `0`.

This can be added before upgrading to 2.2 to ensure the same behavior on first boot after the upgrade. It will result in a harmless cosmetic error in the logs on 2.1.5 since the value does not exist in that version.

If a firewall has more than one LAGG interface configured, enter a tunable for each instance since that is a per-interface option. For `lagg1`, add the following:

```
net.link.lagg.1.lacp.lacp_strict_mode
```

Also with the value set to `0`.

Intel 10Gbit/s ixgbe/ix users with Unsupported SFP modules

The sysctl to allow unsupported SFP modules changed in FreeBSD between the versions used for pfSense 2.1.x and 2.2.

The old tunable was:

```
hw.ixgbe.unsupported_sfp=1
```

This must be changed to:

```
hw.ix.unsupported_sfp=1
```

Edit the setting in `/boot/loader.conf.local` before applying the update and the behavior will be retained.

Layer 7

Layer 7 is deprecated and has been removed. For layer 7 application identification and filtering we recommend using the *Snort IDS/IPS* package with OpenAppID detectors and rules.

Microsoft Load Balancing / Open Mesh Traffic

Windows Network Load Balancing and Open Mesh access points can use multicast MAC address destinations which rely on broken behavior that was incorrectly allowed by default in earlier versions of FreeBSD and pfSense. The fact it worked before was technically a bug, acting in violation of RFC 1812.

> A router MUST not believe any ARP reply that claims that the Link Layer address of another host or router is a broadcast or multicast address.

The default behavior on pfSense 2.2 is correct, but it may be changed.

If this behavior be required, manually add a tunable as follows:

- Navigate to **System > Advanced**, **System Tunables** tab
- Click 
- Enter the following values:
  - **Tunable**: `net.link.ether.inet.allow_multicast`
  - **Description**: Optional. It would be wise to enter the URL to this note or a similar note.
  - **Value**: `1`
- Click **Save**

Upgrading from versions older than pfSense 2.1

See also:

For information about upgrading to current versions, see *Upgrade Guide*.

> **Warning:** Uninstalling **all** packages is **required** when upgrading from old releases. Packages **must be removed** before the upgrade is performed. After the upgrade is complete, packages can be reinstalled. Package configuration is automatically retained.

See the HA section at the end of this document for a High Availability-specific pfsync note about pfSense® software version 2.1 upgrades.

The **State Killing on Gateway Failure** feature (**System > Advanced**, **Miscellaneous tab**) now kills ALL states when a gateway has been detected as down, not only states on the failing WAN. This is done because otherwise the LAN-side states were not killed appropriately, and thus some connections would be in limbo, especially SIP. Due to the change in its behavior, **State Killing on Gateway Failure** is disabled by default in new configurations and is disabled during upgrade to pfSense 2.1.x from 2.0.x or before regardless of the user's previously chosen setting. If the feature is desired even with its new behavior, it must be manually re-enabled post-upgrade.

The **Allow IPv6** checkbox is **NOT** changed on upgrade unlike it was in early pfSense 2.1 BETA snapshots. This was changed so that the user's chosen existing behavior is preserved. To allow IPv6 traffic after an upgrade, the setting must be changed manually. This setting is located on **System > Advanced** on the **Networking** tab. It defaults to allowed for new configurations.

Changes to policy route negation between pfSense 2.0.x and 2.1 may result in local/private traffic hitting policy routing rules that would not have happened on pfSense 2.0.x. This most commonly presents as an inability to reach local networks after upgrading. The automatic policy route negation rules on pfSense 2.0.x were too lenient, and that behavior was corrected. To ensure proper routing to other local interfaces, VPNs, or static route networks rules must be added to the local interfaces to pass traffic to these destinations without a gateway set. And that rule must be above any others that would match and have a gateway set.

Upgrading High Availability Deployments

If upgrading from any previous version of pfSense (1.2.x, 2.0.x, etc) to pfSense 2.1 or later in an HA environment, ensure that the pfsync interface has a rule to pass the correct traffic for state synchronization to work properly. pfSense 2.1 removed the automatic pfsync rule, since the documentation always recommended adding it manually and to add it behind the scenes with no way to block it could be counter-productive and potentially insecure. If the documentation was not followed, and a pfsync or allow all rule was not created on the sync interface, state synchronization may break after this upgrade. Add an appropriate rule to the sync interface and it will work again.

At a minimum, pass traffic of the *pfsync* protocol from a source of the synchronization subnet to all cluster nodes.

Upgrading from versions older than pfSense® 2.0

See also:

For information about upgrading to current versions, see *Upgrade Guide*.

Warning: Uninstalling **all** packages is **required** when upgrading from old releases. Packages **must be removed** before the upgrade is performed. After the upgrade is complete, packages can be reinstalled. Package configuration is automatically retained.

Note for users of the OpenVPN Status Package

If a manual `management` directive was entered into the **Advanced Configuration** of an OpenVPN client or server, it must be removed. The OpenVPN status code is built into pfSense® software version 2.x and later, and it is handled internally. The management directive must be removed or the status of the VPN instance will not be properly reported.

Note for Captive Portal RADIUS WISPr Bandwidth Users

The WISPr RADIUS attributes were incorrectly applied to all versions prior to pfSense 2.0-RELEASE. They were applied as *Kbps* where WISPr is supposed to be *bps*, hence those using WISPr attributes will have one one-thousandth of the previous bandwidth unless the RADIUS server is corrected. The RADIUS server will need to have these values updated to bps for proper functionality once the firewall has been upgraded to pfSense 2.0-RELEASE or later.

International/Special Characters in 1.2.x Configurations

International characters, such as åäö and so on, were not supported on pfSense 1.2.x, but were allowed in some places due to overly lenient input validation and less strict XML parsing. These characters causes invalid XML when they are stored directly, and as such if pfSense 1.2.x did not crash and toss out the configuration with such characters, it will not upgrade to any current version of pfSense software.

pfSense software version 2.0 and later will reset and toss out the config.xml on every reboot if it contains these characters bare, leaving the firewall at an "assign interfaces" prompt since it does not have a valid configuration.

The *config.xml* file can be run through an XML parser such as `xmllint` and the parser will show where problems exist, if any. Fix the errors, and then the corrected configuration can be used for an upgrade. The good news is that these characters are handled properly in most areas of the current pfSense GUI, and they are CDATA escaped so they are safe from such errors.

Upgrading High Availability Deployments

When upgrading from pfSense 1.2.3 to 2.0 or later, Check the CARP VIPs to make sure they are actually on the proper interface. That is, that the interface chosen for the VIP properly matches the subnet in which the CARP VIP resides, and that the subnet mask is proper. pfSense 2.0 validates this more strictly than previous releases, and as a consequence if a CARP VIP was misconfigured on pfSense 1.2.3, it may not upgrade cleanly.

7.6.3 Perform the Upgrade

There are several methods available to update an installation of pfSense® software. Either the WebGUI or the console can be used.

Note: Before performing an upgrade, read through the entire *Upgrade Guide*.

If problems occur during the upgrade process, consult *Troubleshooting Upgrades* for assistance.

Upgrading using the WebGUI

The **Automatic Update** feature contacts a Netgate server and determines if there is a release version newer than the version on the firewall. This check is performed when an administrator visits the dashboard or **System > Update**.

To perform the upgrade in the GUI:

- Navigate to **System > Update** or click  in the System Information dashboard widget next to the new version notification.
- Click  **Confirm** to start the update
- Wait for the upgrade to complete

The update takes a few minutes to download and apply, depending on the speed of the Internet connection being used and the speed of the firewall hardware. The firewall will reboot automatically when finished.

Tip: Monitor the system console during the upgrade if possible to watch for potential problems.

Upgrading using the Console

An update may also be run from the console. The console option is available from any means available for console access: Video/Keyboard, Serial Console, or SSH.

- Connect to the firewall console or login via ssh
- Enter menu option `13`
- Wait for the upgrade to complete

Alternately, from a shell prompt running as root, manually execute the following command:

```
# pfSense-upgrade
```

Tip: When upgrading from SSH, the GNU screen utility can be a useful tool to monitor the upgrade process in environments where the connection to the firewall is unstable:

```
# pkg install screen
# rehash
# screen pfSense-upgrade
```

Reinstalling / Upgrading Configuration

If an upgrade will not function properly on an existing installation, the configuration file can be restored to a freshly installed copy of pfSense software. An older configuration can always be imported into a new version. The upgrade code will make necessary changes to the configuration so it will work with the current version of the software. See *Backup and Recovery* for details.

7.6.4 Upgrading High Availability Clusters

This page provides guidance on upgrading redundant firewalls (CARP, pfsync, XMLRPC config sync) across major versions of pfSense® software. Upgrading from one version to another generally follows the this procedure, exceptions are noted later in the page.

- Review changelog/blog/upgrade guide
- Take a backup from both nodes. **Do not skip this step!**
- Upgrade secondary as described in the *Upgrade Guide*
- Test secondary to be sure it is operating OK – expected packages present, services running, no obvious errors in logs, etc
- Switch CARP to maintenance mode on primary from **Status > CARP**
- Ensure traffic is still flowing properly and that the network is functional. If it is not, then exit maintenance mode on the primary, fix the secondary then try again.
- Upgrade primary as described in the *Upgrade Guide*
- Check primary to ensure it upgraded OK – expected packages present, services running, no obvious errors in logs, etc
- Exit maintenance mode on primary
- Test again

XMLRPC Config Sync Considerations

Upgrade either the primary or the secondary first, leaving the other on the older version until testing is complete.

Supported versions of pfSense software from the last several years properly check for and prevent unintentionally synchronizing data between incompatible versions.

pfsync considerations

The underlying pfsync protocol often changes between FreeBSD versions. Versions of pfSense software with a different base OS version of FreeBSD cannot sync their states between each other. Failover will still function, but not stateful failover so all existing connections will be dropped.

pfsync and Interface-bound States

States contain information about the interface to which they are bound. If the interfaces do not line up on both nodes then the states will not properly sync, for example if WAN is `ix0` on one node and `igb0` on the other.

Adding interfaces to LAGGs can work around this, since then the states would be bound to the lagg on each node rather than the underlying interface. For example, `lagg0` on primary contains `ix0`, `lagg0` on secondary contains `igb0`, but the states are on `lagg0` for both so sync will function.

CARP considerations

CARP is generally the same between versions and will fail over and back as expected.

See also:

- *Troubleshooting Upgrades*

7.6.5 Update Settings

Branch / Tracking Snapshots

By default, the update check looks for officially released versions of pfSense software, but this method can also be used to track development snapshots.

To change the branch used for updates:

- Navigate to **System > Update**
- Set the **Branch** to the desired type of updates
- Wait for the page to refresh and perform a new update check

The branch list will vary depending on the current development cycle. Examples of options that may be found in the list include:

Latest Stable Version Stable versions are the best option, as they see the most testing and are reasonably safe and trouble-free. However, as with any upgrade, read the *changelog* and *update notes* for that release.

Previous Stable Version (Deprecated) A pointer to the previous release so that firewalls may pull packages and update files from the previous release while waiting for a maintenance window or similar upgrade opportunity. May also be labeled "Legacy".

Latest Development Snapshots Tracks development snapshot builds. These may either be snapshots for the next minor or major version depending on the status of the development cycle.

Next Major Version Tracks snapshots for the next major update version. This is riskier, but in some cases may be required for newer hardware or new features that are not yet released. Consult the forum and test in a lab to see if these snapshots are stable in a particular environment.

> **Warning:** Do not run development versions of pfSense software in production environments.

Dashboard Check

The **Dashboard Check** checkbox on **System > Update**, **Update Settings** tab controls whether or not an update check is performed by the **System Information** widget on the dashboard. On firewalls with low resources or slow disks, disabling this check will reduce the load caused by running the check each time an administrator views the dashboard.

GitSync

This section is for developers and should not be used by end users. Leave settings in this area empty or disabled.

See also:

- *Virtualization*
- *Connect to the Console*
- *Troubleshooting Installation Issues*
- *Troubleshooting Upgrades*
- *Troubleshooting Disk and Filesystem Issues*

CHAPTER

EIGHT

CONFIGURATION

8.1 Setup Wizard

The first time a user logs into the pfSense® software GUI, the firewall presents the Setup Wizard automatically. The first page of the wizard is shown in Figure *Setup Wizard Starting Screen*.

Click » **Next** to proceed.

Tip: Using the setup wizard is optional. Click the logo at the top left of the page to exit the wizard at any time.

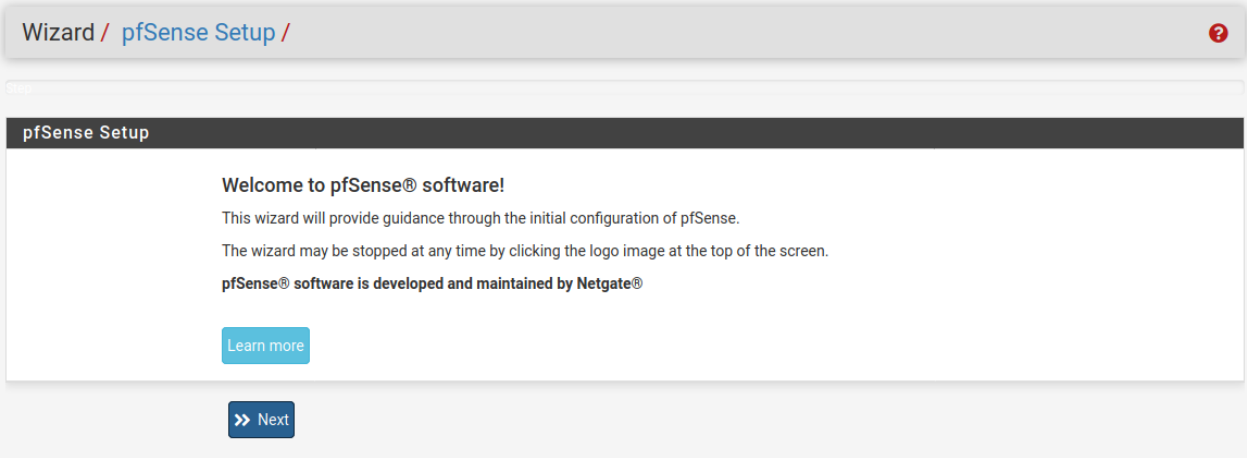


Fig. 1: Setup Wizard Starting Screen

The next screen of the wizard explains the availability of support from Netgate. Click » **Next** again to start the configuration process using the wizard.

8.1.1 General Information Screen

The next screen (Figure *General Information Screen*) configures the name of this firewall, the domain in which it resides, and the DNS servers for the firewall.

Hostname The **Hostname** is a name that should uniquely identify this firewall. It can be nearly anything, but must start with a letter and it may contain only letters, numbers, or a hyphen.

Domain Enter a Domain, e.g. `example.com`. If this network does not have a domain, use `<something>.home.arpa`, where `<something>` is another identifier: a company name, last name, nickname, etc. For example, `company.home.arpa` The hostname and domain name are combined to make up the fully qualified domain name of this firewall.

Primary/Secondary DNS Server The IP address of the Primary DNS Server and Secondary DNS Server, if known.

These DNS servers may be left blank if the DNS Resolver will remain active using its default settings. The default configuration has the DNS Resolver active in resolver mode (not forwarding mode), when set this way, the DNS Resolver does not need forwarding DNS servers as it will communicate directly with Root DNS servers and other authoritative DNS servers. To force the firewall to use these configured DNS servers, enable forwarding mode in the DNS Resolver or use the DNS Forwarder.

If this firewall has a dynamic WAN type such as DHCP, PPTP or PPPoE these may be automatically assigned by the ISP and can be left blank.

Note: The firewall can have more than two DNS servers, add more under **System > General Setup** after completing the wizard.

Override DNS When checked, a dynamic WAN ISP can supply DNS servers which override those set manually. To force the use of only the DNS servers configured manually, uncheck this option.

See also:

For more information on configuring the DNS Resolver, see *DNS Resolver*

Click  **Next** to continue.

8.1.2 NTP and Time Zone Configuration

The next screen (Figure *NTP and Time Zone Setup Screen*) has time-related options.

Time server hostname A Network Time Protocol (NTP) server hostname or IP address. Unless a specific NTP server is required, such as one on LAN, the best practice is to leave the **Time server hostname** at the default `2.pfsense.pool.ntp.org`. This value will pick a set of random servers from a pool of known-good NTP hosts.

To utilize multiple time server pools or individual servers, add them in the same box, separating each server by a space. For example, to use three NTP servers from the pool, enter: `0.pfsense.pool.ntp.org 1.pfsense.pool.ntp.org 2.pfsense.pool.ntp.org`

This numbering is specific to how `.pool.ntp.org` operates and ensures each address is drawn from a unique pool of NTP servers so the same server does not get used twice.

Timezone Choose a geographically named zone which best matches location of this firewall, or any other desired zone.

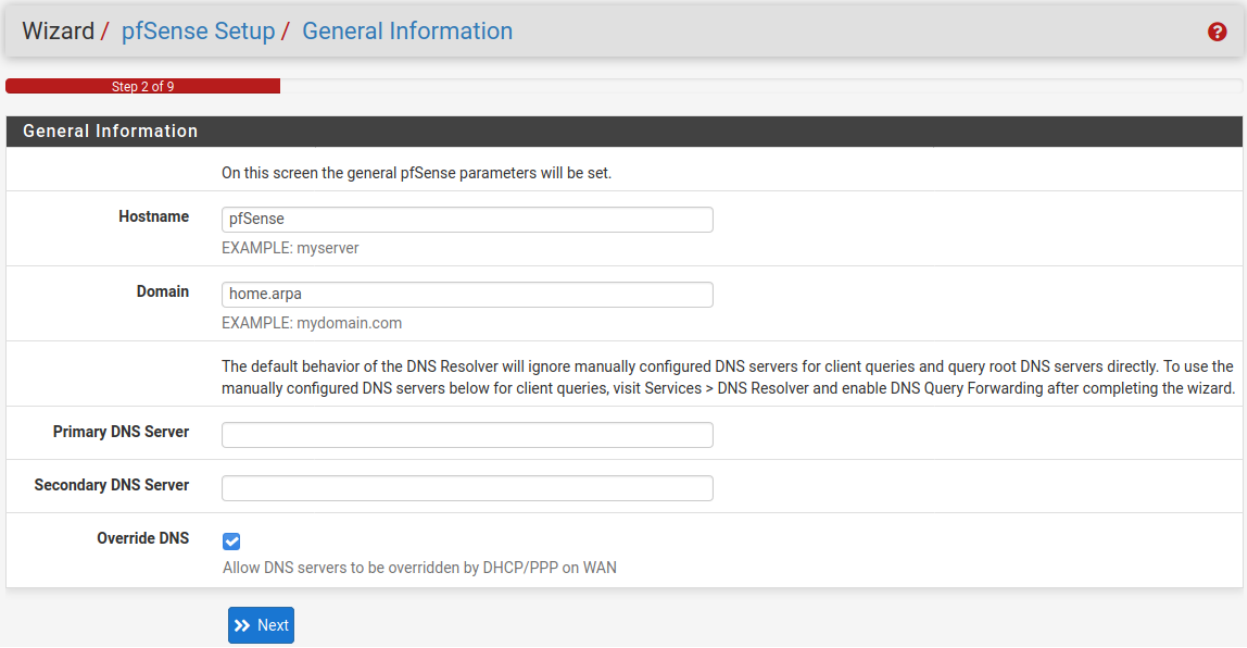


Fig. 2: General Information Screen

Click  **Next** to continue.

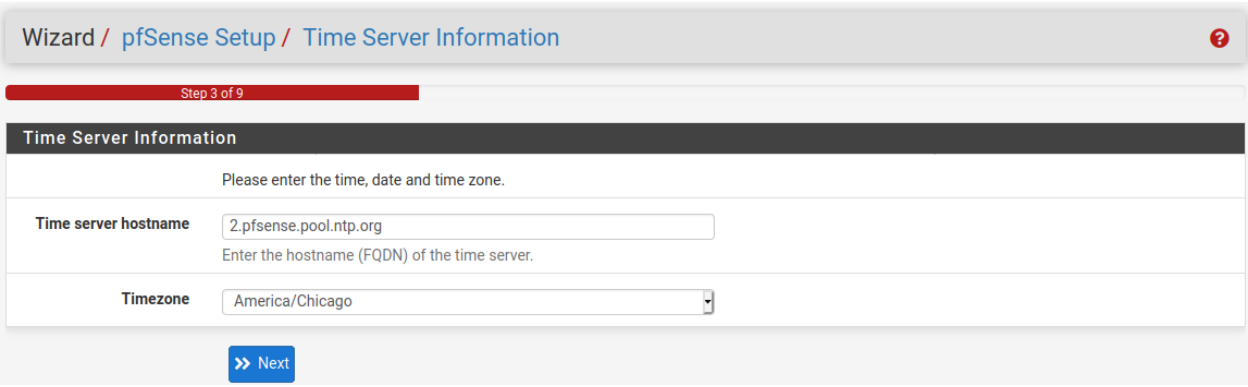


Fig. 3: NTP and Time Zone Setup Screen

8.1.3 WAN Configuration

The next page of the wizard configures the WAN interface of the firewall. This is the external network facing the ISP or upstream router, so the wizard offers configuration choices to support several common ISP connection types.

WAN Type The **Selected Type** (Figure *WAN Configuration*) must match the type of WAN required by the ISP, or whatever the previous firewall or router was configured to use. Possible choices are *Static*, *DHCP*, *PPPoE*, and *PPTP*. The default choice is *DHCP* due to the fact that it is the most common, and for the majority of cases this setting allows a firewall to "Just Work" without additional configuration. If the WAN type is not known, or specific settings for the WAN are not known, this information must be obtained from the ISP. If the required WAN type is not available in the wizard,

or to read more information about the different WAN types, see *Interface Types and Configuration*.

Note: If the WAN interface is wireless, additional options will be presented by the wizard which are not covered during this walkthrough of the standard Setup Wizard. Refer to *Wireless*, which has a section on Wireless WAN for additional information. If any of the options are unclear, skip the WAN setup for now, and then perform the wireless configuration afterward.



Fig. 4: WAN Configuration

MAC Address This field, shown in Figure *General WAN Configuration*, changes the MAC address used on the WAN network interface. This is also known as "spoofing" the MAC address.

Note: The problems alleviated by spoofing a MAC address are typically temporary and easily worked around. The best course of action is to maintain the original hardware MAC address, resorting to spoofing only when absolutely necessary.

Changing the MAC address can be useful when replacing an existing piece of network equipment. Certain ISPs, primarily Cable providers, will not work properly if a new MAC address is encountered. Some Internet providers require power cycling the modem, others require registering the new address over the phone. Additionally, if this WAN connection is on a network segment with other systems that locate it via ARP, changing the MAC to match and older piece of equipment may also help ease the transition, rather than having to clear ARP caches or update static ARP entries.

Warning: If this firewall will ever be used as part of a *High Availability Cluster*, do not spoof the MAC address.

Maximum Transmission Unit (MTU) The MTU field, shown in Figure *General WAN Configuration*, can typically be left blank, but can be changed when necessary. Some situations may call for a lower MTU to ensure packets are sized appropriately for an Internet connection. In most cases, the default assumed values for the WAN connection type will work properly.

Maximum Segment Size (MSS) MSS, shown in Figure *General WAN Configuration* can typically be left blank, but can be changed when necessary. This field enables MSS clamping, which ensures TCP packet sizes remain adequately small for a particular Internet connection.

Static IP Configuration If the "Static" choice for the WAN type is selected, the **IP address**, **Subnet Mask**, and **Upstream Gateway** must all be filled in (Figure *Static IP Settings*). This information must be obtained from the ISP or whoever controls the network on the WAN side of this firewall. The **IP Address** and **Upstream Gateway** must both reside in the same Subnet.

DHCP Hostname This field (Figure *DHCP Hostname Setting*) is only required by a few ISPs. This value is sent along with the DHCP request to obtain a WAN IP address. If the value for this field is unknown, try leaving it blank unless directed otherwise by the ISP.

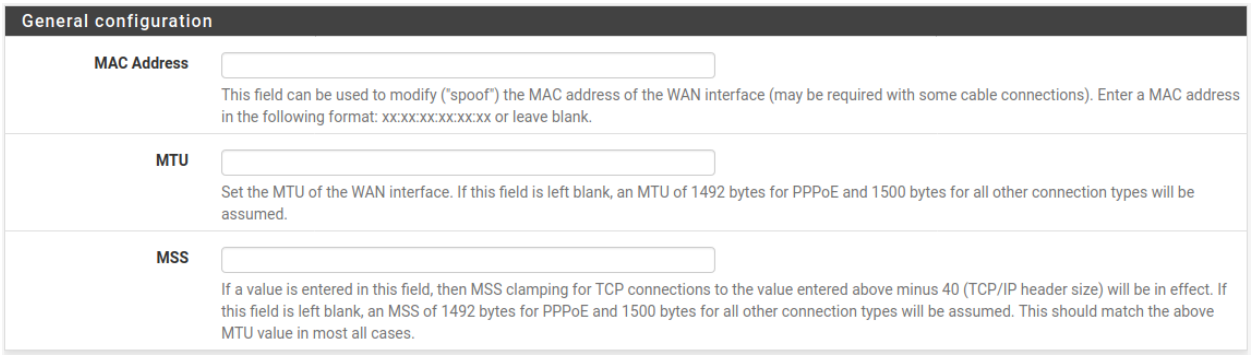


Fig. 5: General WAN Configuration

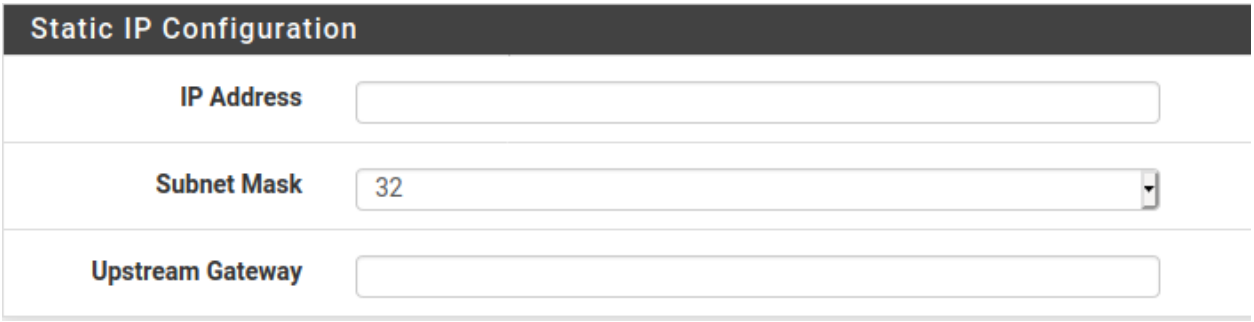


Fig. 6: Static IP Settings

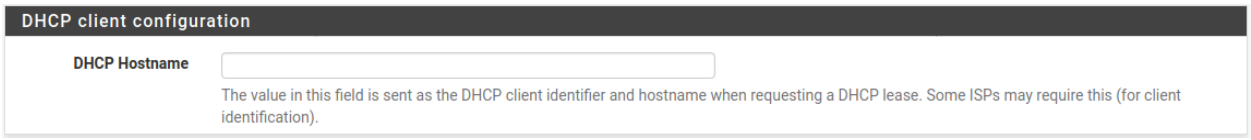


Fig. 7: DHCP Hostname Setting

PPPoE Configuration When using the PPPoE (Point-to-Point Protocol over Ethernet) WAN type (Figure *PPPoE Configuration*), The **PPPoE Username** and **PPPoE Password** fields are required, at a minimum. The values for these fields are determined by the ISP.

PPPoE Username The login name for PPPoE authentication. The format is controlled by the ISP, but commonly uses an e-mail address style such as `myname@example.com`.

PPPoE Password The password to login to the account specified by the username above. The password is masked by default. To view the entered password, check **Reveal password characters**.

PPPoE Service Name The PPPoE Service name may be required by an ISP, but is typically left blank. When in doubt, leave it blank or contact the ISP and ask if it is necessary.

PPPoE Dial on Demand This option leaves the connection down/offline until data is requested that would need the connection to the Internet. PPPoE logins happen quite fast, so in most cases the delay while the connection is setup would be negligible. If public services are hosted behind this firewall, do not check this option as an online connection must be maintained as much as possible in that case. Also note that this choice will not drop an existing connection.

PPPoE Idle Timeout Specifies how much time the PPPoE connection remain up without transmitting data before disconnecting. This is only useful when coupled with Dial on demand, and is typically left blank (disabled).

Note: This option also requires the deactivation of gateway monitoring, otherwise the connection will never be idle.

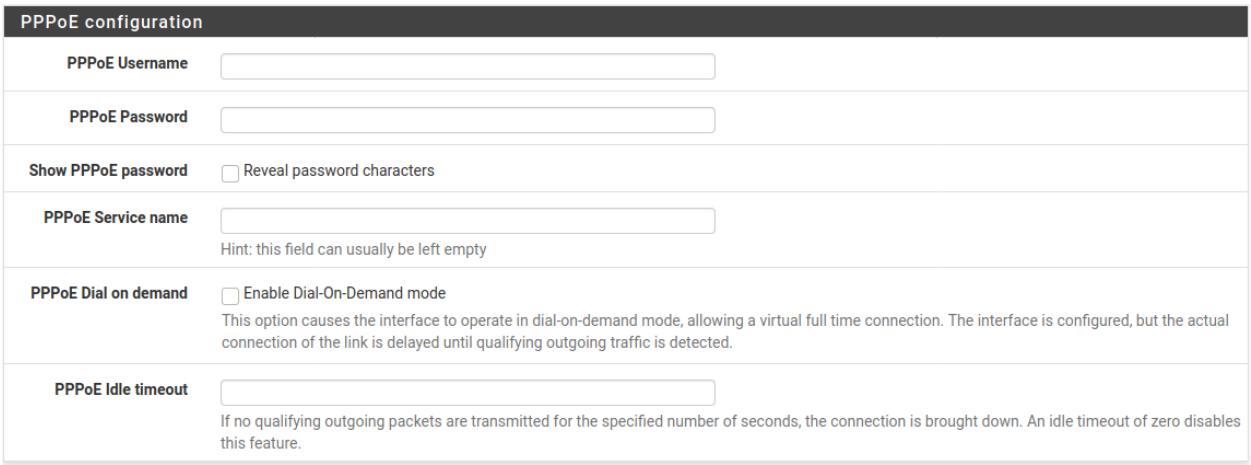


Fig. 8: PPPoE Configuration

PPTP Configuration The PPTP (Point-to-Point Tunneling Protocol) WAN type (Figure *PPTP WAN Configuration*) is for ISPs that require a PPTP login, **not** for connecting to a remote PPTP VPN. These settings, much like the PPPoE settings, will be provided by the ISP. A few additional options are required:

Local IP Address The local (usually private) address used by this firewall to establish the PPTP connection.

CIDR Subnet Mask The subnet mask for the local address.

Remote IP Address The PPTP server address, which is usually inside the same subnet as the **Local IP address**.

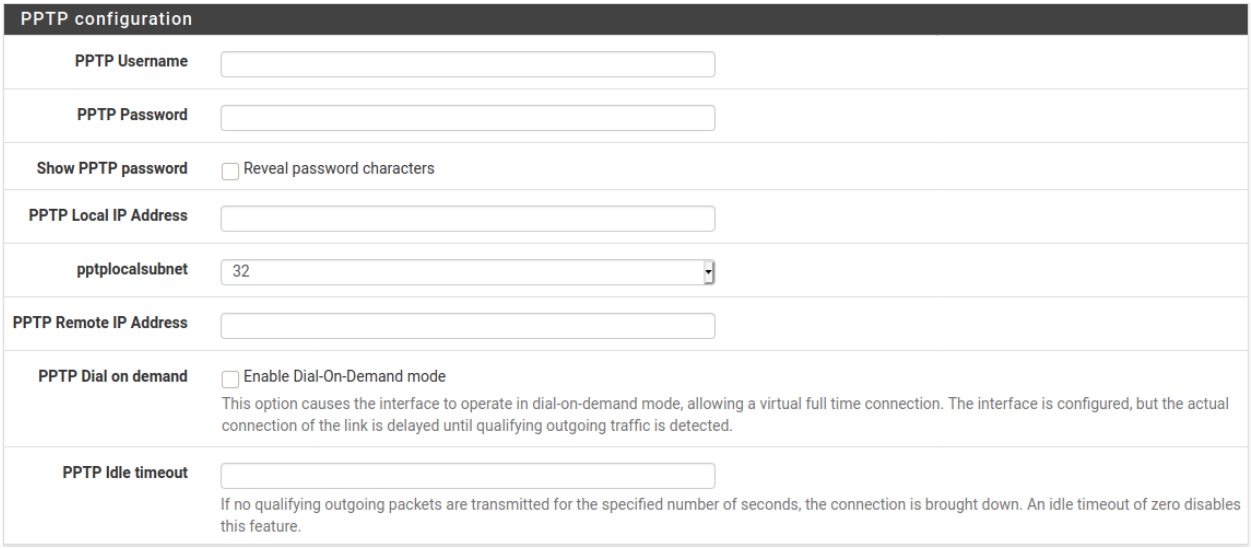


Fig. 9: PPTP WAN Configuration

These last two options, seen in Figure *Built-in Ingress Filtering Options*, are useful for preventing invalid traffic from entering the network protected by this firewall, also known as "Ingress Filtering".

Block RFC 1918 Private Networks Blocks connections sourced from registered private networks such as `192.168.x.x` and `10.x.x.x` attempting to enter the WAN interface . A full list of these networks is in *Private IP Addresses*.

Block Bogon Networks When active, the firewall blocks traffic from entering if it is sourced from reserved or unassigned IP space that should not be in use. The list of bogon networks is updated periodically in the background, and requires no manual maintenance. Bogon networks are further explained in *Block Bogon Networks*.

Click  **Next** to continue once the WAN settings have been filled in.

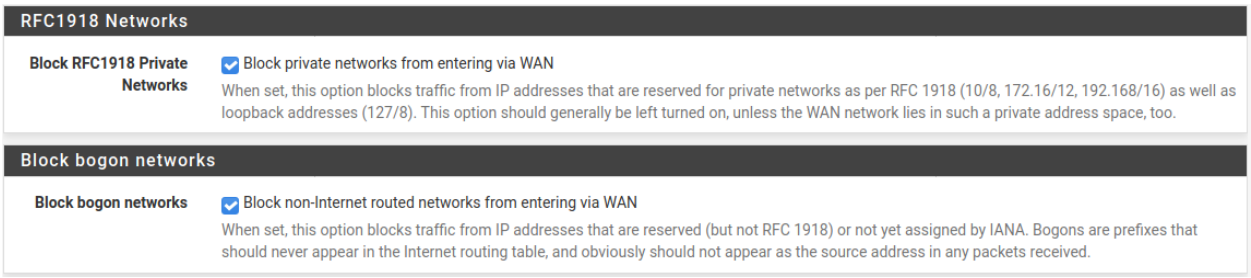


Fig. 10: Built-in Ingress Filtering Options

8.1.4 LAN Interface Configuration

This page of the wizard configures the **LAN IP Address** and **Subnet Mask** (Figure *LAN Configuration*).

If this firewall will not connect to any other network via VPN, the default `192.168.1.0/24` network may be acceptable. If this network must be connected to another network, including via VPN from remote locations, choose a private IP address range much more obscure than the common default of `192.168.1.0/24`. IP space within the `172.16.0.0/12` RFC 1918 private address block is generally the least frequently used, so choose something between `172.16.x.x` and `172.31.x.x` to help avoid VPN connectivity difficulties.

If the LAN is `192.168.1.x` and a remote client is at a wireless hotspot using `192.168.1.x` (very common), the client will not be able to communicate across the VPN. In that case, `192.168.1.x` is the local network for the client at the hotspot, not the remote network over the VPN.

If the **LAN IP Address** must be changed, enter it here along with a new **Subnet Mask**. If these settings are changed, the IP address of the computer used to complete the wizard must also be changed if it is connected through the LAN. Release/renew its DHCP lease, or perform a "Repair" or "Diagnose" on the network interface when finished with the setup wizard.

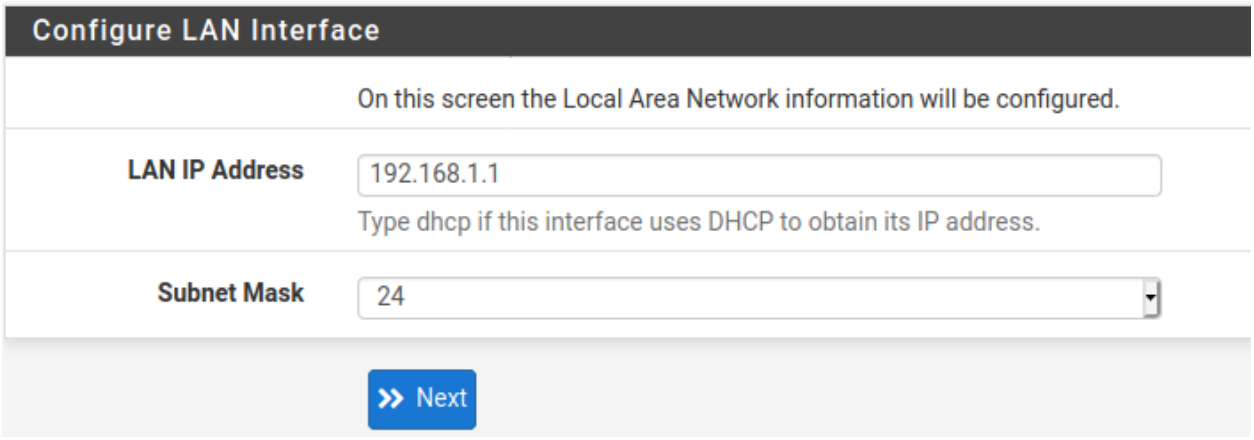


Fig. 11: LAN Configuration

Click » **Next** to continue.

8.1.5 Set admin password

Next, change the administrative password for the GUI as shown in Figure *Change Administrative Password*. The best practice is to use a strong and secure password, but no restrictions are automatically enforced. Enter the password in the **Admin Password** and confirmation box to be sure that has been entered correctly.

Click » **Next** to continue.

Warning: Do not leave the password set to the default `pfsense`. If access to the firewall administration via GUI or SSH is exposed to the Internet, intentionally or accidentally, the firewall could easily be compromised if it still uses the default password.

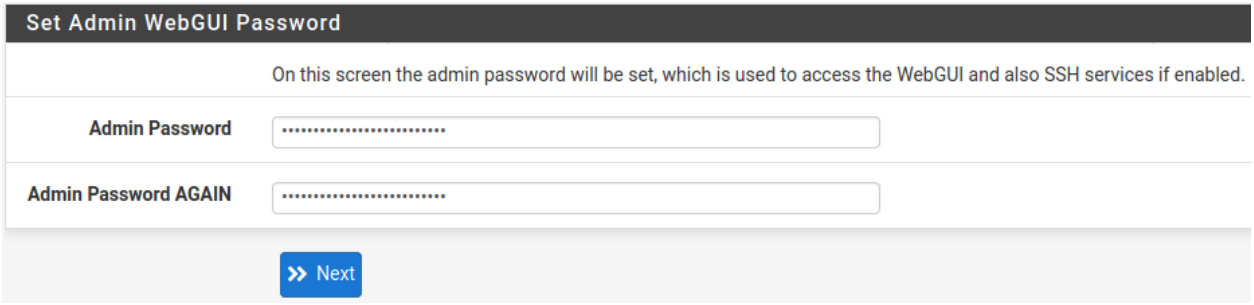


Fig. 12: Change Administrative Password

8.1.6 Completing the Setup Wizard

That completes the setup wizard configuration. Click **Reload** (Figure *Reload the GUI*) and the GUI will apply the settings from the wizard and reload services changed by the wizard.

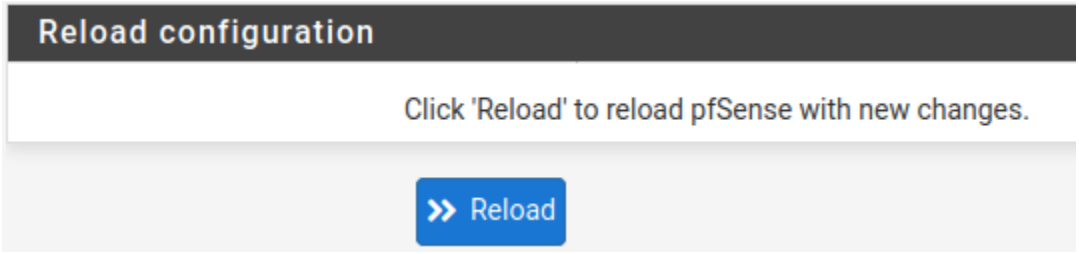


Fig. 13: Reload the GUI

Tip: If the LAN IP address was changed in the wizard and the wizard was run from the LAN, adjust the client computer's IP address accordingly after clicking **Reload**.

When prompted to login again, enter the new password. The username remains `admin`.

After reloading, the final screen of the wizard includes convenient links to check for updates, get support, and other resources. Click **Finish** to complete and exit the wizard.

At this point the firewall will have basic connectivity to the Internet via the WAN and clients on the LAN side will be able to reach Internet sites through this firewall.

If at any time this initial configuration must be repeated, revisit the wizard at **System > Setup Wizard** from within the GUI.

8.2 Interface Configuration

Basic aspects of interface configuration within pfSense® software can be performed at the console and in the setup wizard to start, but changes may also be made after the initial setup by visiting pages under the **Interfaces** menu. A few basics are covered here, the details can be found in *Interface Types and Configuration*.

8.2.1 Assign interfaces

Interfaces added after the initial setup may be assigned roles by visiting **Interfaces > Assignments**. There are numerous tabs on that page used for assigning and creating different types of interfaces. The two most commonly used tabs are **Interface assignments** and **VLANs**.

See also:

VLAN configuration is covered in *Virtual LANs (VLANs)*.

The **Interface assignments** tab shows a list of all currently assigned interfaces: WAN, LAN, and any OPTx entries configured on the firewall. Next to each interface is a drop-down list of all network interfaces/ports found on the system. This list includes hardware interfaces as well as VLAN interfaces and other virtual interface types. The MAC address, VLAN tag, or other identifying information is printed along side the interface name to aid in identification.

The other tabs, much like the **VLAN** tab, are there to create additional interfaces which can then be assigned. All of these interface types are covered in *Interface Types and Configuration*.

To change an existing interface assignment to another network port:

- Navigate to **Interfaces > Assignments**
- Locate the interface to change in the list
- Select the new network port from the drop-down list on the row for that interface
- Click **Save**

To add a new interface from the list of unused network ports:

- Navigate to **Interfaces > Assignments**
- Select the port to use from the drop-down list labeled **Available Network Ports**
- Click  **Add**

This action will add another line with a new OPT interface numbered higher than any existing OPT interface, or if this is the first additional interface, *OPT1*.

8.2.2 Interface Configuration Basics

Interfaces are configured by choosing their entry from under the **Interfaces** menu. For example, to configure the WAN interface, choose **Interfaces > WAN**.

Every interface is configured in the same manner and any interface can be configured as any interface type (Static, DHCP, PPPoE, etc). Additionally, the blocking of private networks and bogon networks may be performed on any interface. Every interface can be renamed, including WAN and LAN, to a custom name. Furthermore, every interface can be enabled and disabled as desired, so long as a minimum of one interface remains enabled.

See also:

For detailed interface configuration information, see *Interface Types and Configuration*

The **IPv4 Configuration Type** can be changed between *Static IPv4*, *DHCP*, *PPPoE*, *PPP*, *PPTP*, *L2TP*, or *None* to leave the interface without an IPv4 address. When *Static IPv4* is used, an **IPv4 Address**, subnet mask, and **IPv4 Upstream Gateway** may be set. If one of the other options is chosen, then type-specific fields appear to configure each type.

The **IPv6 Configuration Type** can be set to *Static IPv6*, *DHCP6*, *SLAAC*, *6rd Tunnel*, *6to4 Tunnel*, *Track Interface*, or *None* to leave IPv6 unconfigured on the interface. When Static IPv6 is selected, set an **IPv6 address**, prefix length, and **IPv6 Upstream Gateway**.

If this a wireless interface, the page will contain many additional options to configure the wireless portion of the interface. Consult *Wireless* for details.

Note: Selecting a **Gateway** from the drop-down list, or adding a new gateway and selecting it, will direct the firewall to treat this interface as a WAN type interface for NAT and related functions. This is not desirable for internal-facing interfaces such as LAN or a DMZ. Gateways may still be utilized on those interfaces for static routes and other purposes *without* selecting a **Gateway** here on the interfaces page.

8.3 Managing Lists in the GUI

The pfSense® software GUI has a common set of icons which are used for managing lists and collections of objects throughout the firewall. Not every icon is used in every page, but their meanings are consistent based on the context in which they are seen. Examples of such lists include firewall rules, NAT rules, IPsec, OpenVPN, and certificates.

 Add a new item to a list

 Add an item to the beginning of a list

 Add an item to the end of a list

 Edit an existing item

 Copy an item (create a new item based on the selected item)

 Disable an active item

 Enable a disabled item

 Delete an item

 Used for moving entries after selecting one or more items. Click to move the selected items above this row. Shift-click to move the selected items below this row.

Sections may have their own icons specific to each area. Consult the appropriate sections of this documentation for specifics about icons found in other parts of the firewall.

Tip: To determine which action an icon will perform, hover over the icon with the mouse pointer and a tooltip will display a short description of the icon's purpose.

8.4 Quickly Navigate the GUI with Shortcuts

Many areas of the GUI have shortcut icons present in the area known as the "Breadcrumb Bar", as seen in Figure *Shortcuts Example*. These shortcut icons reduce the amount of hunting required to locate related pages, allowing a firewall administrator to navigate quickly between the status page of a service, its logs, and configuration. The shortcuts for a given topic are present on each page related to that topic.

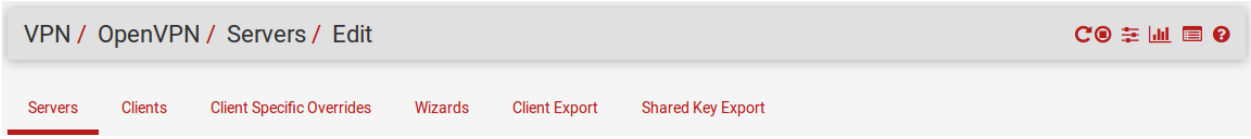


Fig. 14: Shortcuts Example

Note: Shortcut icons only appear when their respective actions are possible and the target pages exist. Not every section has every icon.

The shortcut icons have the following effects when they appear in the GUI:


Start Service If the service is stopped, this icon starts the service.


Restart Service If the service is running, this icon restarts the service.

> **Note:** Some services will stop and start, others reload the configuration. Check the documentation of each service for details.


Stop Service If the service is running, this icon stops the service.


Related Settings This icon navigates to the **settings** page for this section.


Status Page Link This icon navigates to the the **status** page for this section.


Log Page Link This icon navigates to the the **logs** page for this section.


Help Link This icon navigates to a related help topic for this page.

The *Service Status* page (**Status > Services**) also has shortcut controls for pages related to each service, as shown in Figure *Shortcuts on Service Status*. The icons have the same meaning as in the above section.

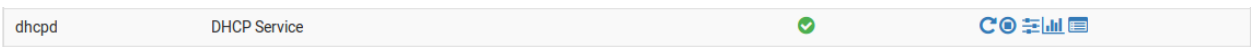


Fig. 15: Shortcuts on Service Status

8.5 General Configuration Options

System > General Setup contains options which set basic configuration items for pfSense® software. A few of these options are also found in the *Setup Wizard*.

Hostname The **Hostname** is the short name for this firewall, such as `firewall1`, `hq-fw`, or `site1`. The name must start with a letter and it may contain only letters, numbers, or a hyphen.

Domain The **Domain** name for this firewall, e.g. `example.com` . If this network does not have a domain, use `<something>.localdomain`, where `<something>` is another identifier: a company name, last name, nickname, etc. For example, `company.localdomain`

The **Hostname** and **Domain** name are combined to make up the Fully Qualified Domain Name (FQDN) of this firewall. For example, if the **Hostname** is `fw1` and the **Domain** is `example.com`, then the FQDN is `fw1.example.com`.

8.5.1 DNS Server Settings

Options in this section control how the firewall resolves hostnames using DNS.

Note: The DNS Resolver is active by default and uses **resolver** mode (*DNS Resolver*). When set this way the DNS Resolver does not need forwarding DNS servers as it will communicate directly with root DNS servers and other authoritative DNS servers.

To use the servers in this list, switch the DNS resolver to forwarding mode. The DNS Forwarder (*DNS Forwarder*) only supports forwarding mode and will always use the servers from this list.

DNS Servers

This page supports multiple DNS servers managed as a list. To add more DNS servers, click  **Add DNS Server**.

To remove an entry from the list click  **Delete**.

The DNS server list may be left blank if the DNS Resolver will remain active using its default settings. If this firewall has a dynamic WAN type such as DHCP, PPTP or PPPoE these may be automatically assigned by the ISP and can also be left blank.

Each DNS server entry has the following properties:

DNS Server Address The IP address of the DNS Server.

DNS Server Hostname The FQDN of the DNS server, used to validate DNS server certificates when using DNS over TLS (*DNS Resolver*).

DNS Server Gateway The gateway through which the firewall will reach this DNS server.

This is useful in a Multi-WAN scenario where, ideally, the firewall will have at least one DNS server configured per WAN. More information on DNS for Multi-WAN can be found in *DNS Servers and Static Routes*.

DNS Resolution Behavior

These options fine tune the way the firewall utilizes DNS servers.

DNS Server Override When checked, a dynamic WAN ISP can supply DNS servers which override those set manually. To force the use of only the DNS servers on this page, uncheck this option. This does not apply to the DNS Resolver when acting in resolver mode.

Disable DNS Forwarder By default, the firewall will consult the DNS Resolver or DNS Forwarder running on this firewall to resolve hostnames for itself. It does this by listing localhost (`127.0.0.1`) as its first DNS server internally. Activating this option disables this behavior, forcing the firewall to use the DNS servers configured above instead of itself.

8.5.2 Localization

Options in this section control the firewall's clock display and language.

Time Zone The time zone used by the firewall for its clock. Choose a geographically named zone which best matches location of this firewall, or a common zone such as UTC. The firewall clock, log entries, and other areas of the firewall base their time on this zone. Changing the zone requires a reboot to fully activate the new zone in all areas of the firewall.

Time Servers *Network Time Protocol (NTP)* server hostnames or IP addresses. Unless a specific NTP server is required, such as one on LAN, the best practice is to leave the **Time Servers** value at the default `0.pfsense.pool.ntp.org`. This value will pick random servers from a pool of known-good NTP hosts.

To utilize multiple time servers or pools, add them in the same box, separating each entry by a space. For example, to use three NTP servers from the pool, enter:

```
0.pfsense.pool.ntp.org 1.pfsense.pool.ntp.org 2.pfsense.pool.ntp.org
```

This numbering is specific to how `.pool.ntp.org` operates and ensures each address is drawn from a unique pool of NTP servers so the same server does not get used twice.

Language The language used by the GUI. The GUI has been translated into multiple languages in addition to the default *English* language.

8.5.3 webConfigurator

Options in this section control various aspects of GUI behavior.

Theme The **Theme** controls the look and feel of the GUI. Several themes are included in the base system, and they only make cosmetic not functional changes to the WebGUI.

Top Navigation This option controls the behavior of the menu bar at the top of each page. There are two possible choices:

Scrolls with page The default behavior. When the page scrolls, the navigation remains at the top of the *page*, so it is no longer visible as it scrolls off the top of the window. This is the best option for most situations.

Fixed When selected, the navigation remains fixed at the top of the *window*, always visible and available for use. This behavior can be convenient, but on smaller screens such as tablets and mobile devices, long menus can be cut off, leaving options at the bottom unreachable.

Hostname in Menu When set, the GUI includes the firewall **Hostname** or **Fully Qualified Domain Name** in the menu bar for reference. This can aid when maintaining multiple firewalls, making it easier to distinguish them without looking at the browser title or tab text.

Dashboard Columns The dashboard is limited to `2` columns by default. On wider displays, additional columns can make better use of horizontal screen space. The maximum number of columns is `4`.

Interfaces Sort When unset (default), the GUI presents interfaces in their natural order from the configuration. This is critical for functions such as High Availability which require specific interface ordering. When this option is set, the GUI sorts the interface list alphabetically.

Associated Panels Show/Hide Some GUI pages contain collapsible panels with settings or functions. These panels take up extra screen space, so they are hidden by default. For firewall administrators who use the panels frequently, this can be slow and inefficient. The options in this group make the GUI show these panels by default instead of hiding them.

Available Widgets Controls the **Available Widgets** panel on the Dashboard.

Log Filter Controls the log filtering () panel used for searching log entries under **Status > System Logs**.

Manage Log Controls the per-log settings in the **Manage Log** () panel available for each log under **Status > System Logs**.

Monitoring Settings Controls the options panel used to change the graphs at **Status > Monitoring**.

Require State Filter When set, the state table contents at **Diagnostics > States** are suppressed by the GUI unless a filter string is present. This helps the GUI handle large state tables which otherwise may fail to load.

Left Column Labels When checked, the option labels in the left column are set to toggle options when clicked. This can be convenient if the firewall administrator is used to the behavior, but it can also be problematic on mobile or in cases when the behavior is unexpected.

Alias Popups When set, the tooltip presented by the GUI when hovering over an alias in a rule list only shows the alias description. When unset, the contents of the alias are included in the tooltip. For firewalls with large aliases, this may cause performance or browser rendering issues.

Disable Dragging When set, the GUI disables drag-and-drop on rule lists. Most users find drag-and-drop to be convenient and beneficial, thus the feature is enabled by default. Users who find the behavior undesirable can set this option.

Login Page Color Controls the color of the login page, which is independent of the theme.

Login Hostname When set, the GUI includes the hostname on the login form. This can be considered a security risk since it exposes information about the firewall to users who have not yet authenticated. If the firewall GUI is only reachable by authorized management clients, the convenience may outweigh the potential risk.

8.6 Advanced Configuration Options

System > Advanced contains numerous options of an advanced nature. These options customize the firewall behavior for more complex environments. Most administrators will not need to adjust these options for basic deployments.

Some of these options are covered in more detail in other sections of the documentation where their discussion is more topical or relevant, but they are all mentioned here with a brief description.

8.6.1 Admin Access Tab

The options on the **Admin Access** tab govern various methods for administering the firewall, including via the web interface, SSH, serial, and physical console.

webConfigurator (GUI)

Protocol

The protocol used by the GUI to accept web browser connections. May be one of:

HTTP Plain unencrypted HTTP. Insecure and basic, but widely compatible and less likely to have client issues. Should not be used in most cases, and should never be exposed to insecure networks.

HTTPS (SSL/TLS) Encrypted ("Secure") HTTP. Protects communication between the client browser and the firewall GUI. Requires an **SSL/TLS certificate** to function. Depending on the browser and certificate configuration, there may be compatibility issues, but typically these are easily overcome by using current versions.

Note: The best practice is to use HTTPS so only encrypted traffic is exchanged between the GUI and clients.

SSL/TLS Certificate

The **SSL/TLS Certificate** to be used by the GUI in **HTTPS (SSL/TLS)** mode.

The firewall automatically generates a default self-signed certificate on the first boot. That is not an ideal situation, but is better than no encryption at all.

The primary disadvantage of a self-signed certificate is the lack of assurance of the identity of the host, since the certificate is not signed by a Certificate Authority trusted by the browser. Additionally, because for the bulk of Internet users such an invalid certificate should be considered a risk, modern browsers may restrict how such certificates are handled. Firefox, for example, gives a warning screen and forces the user to import the certificate and allow a permanent exception. Chrome shows a warning screen with a link to continue.

Tip: To use an externally signed SSL certificate and key, import them using the *Certificate Manager*, then select the certificate here.

Tip: The *ACME Package* can utilize the free Let's Encrypt service to automatically obtain and update a signed certificate for the GUI or for other purposes on the firewall.

Tip: Another alternate technique is to generate a self-signed CA and then generate a GUI certificate from that CA. Export the CA from the firewall and then import that CA into client browsers manually. Using this method, all certificates signed by that CA will be trusted by browsers. Specifics vary by client platform.

Tip: To generate a new self-signed certificate for the GUI, connect using the console or ssh and from a shell prompt, run the following command:

```
# pfSsh.php playback generateguicert
```

TCP Port

The port used by the GUI for accepting connections from browsers. By default the GUI uses HTTPS on port `443` with a redirect from port `80` for the best compatibility and ease of initial configuration. To change the port, enter a new port number into the **TCP Port** field.

Note: Moving the WebGUI to an alternate port is preferred by some administrators for security by obscurity reasons, though such practices should not be considered as offering any security benefit. Do not expose the GUI to untrusted networks such as the Internet.

Tip: Moving the GUI to another port will free up the standard web ports for use with port forwards or other services such as HAproxy.

Max Processes

The number of web server worker proceses used by the GUI when listening for client browser connections. The default value is `2`.

If multiple administrators view the GUI at the same time and pages are taking too long to load, or are failing to load, then increase the **Max Processes** value.

WebGUI Redirect

Controls whether or not the firewall runs a redirect on port `80` so that if a browser attempts to access the firewall with HTTP, the firewall will accept the request and then redirect the browser to the **TCP Port** used by the GUI (e.g. HTTPS on port `443`).

The redirect is enabled by default for ease of access and compatibility.

Disabling the redirect allows another daemon to bind to port `80`.

HSTS

Controls whether the GUI web server sends the `Strict-Transport-Security` HTTPS response header (HSTS) to the browser. Check this box to disable the behavior.

HSTS forces the browser to use only HTTPS for future requests to the firewall FQDN to ensure it does not accidentally downgrade to an unencrypted connection.

Warning: When disabling HSTS, clients which visited the GUI when HSTS was enabled must perform browser-specific steps for the change to take effect. Consult browser documentation for information on clearing cached HSTS behavior.

OCSP Must-Staple

Controls whether or not the GUI web server forcefully enables OCSP Stapling.

If the GUI **SSL/TLS Certificate** requires OCSP Stapling, this behavior is automatically enabled by the GUI web server. If the certificate property cannot be automatically determined by the firewall, this option can force the behavior.

Tip: Import the full CA and certificate chain or this option will be ignored by the GUI web server.

WebGUI Login Autocomplete

Controls whether or not the login form allows autocomplete so browsers can save the login credentials, for convenience.

In high-security environments, such as those that must adhere to specific security compliance standards, this behavior is not acceptable.

Note: This only controls autocomplete on the login form.

Warning: Few modern browsers respect this option. Many still offer to save passwords even when the form specifies that the browser must not allow the behavior. This behavior must be controlled or changed using browser options.

WebGUI login messages

Controls whether or not the firewall prints successful login messages to the console and system log.

On hardware with a PC speaker, these console messages generate a beep from the speaker, which some users find undesirable.

Checking this option stops the log message and the resulting beep.

Anti-lockout

Controls whether or not the firewall adds special rules to permit access to the WebGUI port and SSH port on the LAN interface by default.

These special rules override user-defined filter rules and prevent the user from accidentally locking themselves out of the firewall GUI or SSH. To control which LAN IP addresses may access the GUI and SSH using firewall rules, disable the anti-lockout rules.

When two or more interfaces are present, the firewall puts anti-lockout rules on the LAN interface; If only one interface is configured, the firewall places rules on that interface instead.

Warning: Filter rules must be in place to allow GUI access before enabling this option! If the LAN rules do not allow access to the GUI, removing the anti-lockout rule will block access to the GUI, potentially leaving the administrator without a means to reach the firewall.

Note: Resetting the LAN IP address from the console also resets the anti-lockout rule. If administrative access is unavailable after enabling this option, choose the console menu option `2`, then choose to set the LAN IP address, and enter in the exact same IP address and accompanying information.

DNS Rebind Check

Controls whether or not the DNS resolver or forwarder performs DNS rebinding checks. These checks prevent the firewall from receiving DNS responses containing private IP addresses from DNS servers to prevent DNS rebinding attacks.

Note: When accessing the firewall by IP address, these checks are not enforced because the attack is only relevant when using a hostname.

Check this box to disable DNS rebinding protection if it interferes with GUI access or name resolution.

See also:

More detail on DNS rebinding attacks may be found on Wikipedia.

The most common case for disabling DNS rebinding checks is when the firewall is set to use an internal DNS server which will return private (RFC1918) answers for hostnames.

Tip: Instead of disabling all DNS rebinding protections, the checks can be selectively disabled on a per-domain basis in the DNS Resolver or DNS Forwarder. See *DNS Resolver and DNS Rebinding Protection* and *DNS Forwarder and DNS Rebinding Protection*.

Browser HTTP_REFERER enforcement

Controls whether or not the GUI checks and enforces HTTP_REFERER contents.

The GUI checks the referring URL sent by a client browser to ensure that the form was submitted from this firewall. This check prevents a form on another site from submitting a request to the firewall, changing an option when the administrator did not intend for that to happen.

This also breaks some convenience behaviors, such as having a page that links to various firewall devices, though the benefits of the check typically outweigh the advantage of those behaviors.

Alternate Hostnames

A list of **Alternate Hostnames** for the firewall allowed by **DNS Rebind Checks** and **HTTP_REFERER Enforcement**. To keep these features active, but alter their behavior slightly, add **Alternate Hostnames**.

By default the GUI allows access to the hostname configured on the firewall and all IP addresses configured on the firewall. Hostnames in this field are allowed by the firewall for GUI access and for referring URL purposes.

Man-In-The-Middle Attack/Warning

If a browser attempts to access the GUI using an IP address that is not configured on the firewall, such as a port forward from another firewall, the GUI prints a message indicating that access to the firewall may be compromised due to a Man-In-The-Middle (MITM) attack.

If such a forwarding was deliberatey configured on this firewall or on a firewall upstream, the message may be safely ignored. If access to the firewall should have been direct, then take great care before logging in to ensure the login credentials are not being routed through an untrusted system.

Access is not disabled by the firewall in this case, it only prints a warning, so there is no option to disable this behavior.

Browser Tab Text

By default, the GUI prints the firewall hostname first in the page/tab title, followed by the page name. To reverse this behavior and show the page name first and hostname second, check **Display page name first in browser tab**.

Administrators who access many firewalls at the same time in separate tabs tend to prefer having the hostname first (default). Administrators who access one firewall with many pages in separate tabs tend to prefer having the page name first.

Secure Shell (SSH)

The Secure Shell (SSH) server provides remote console access and file management. A user can connect with any standard SSH client, such as the OpenSSH command line ssh client, PuTTY, SecureCRT, or iTerm2.

When using SSH, both the `admin` username and `root` username are accessible using the admin account credentials.

Users in the User Manager that have the `User - System - Shell account access` privilege are also allowed to login over ssh. These users do not have root access privileges, and do not print the menu when they login because many of the options require root privileges.

Tip: To grant users additional shell privileges, use the *sudo package*.

File transfers to and from the firewall are also possible by using a Secure Copy (SCP) client such as the OpenSSH command line `scp`, FileZilla, WinSCP or Fugu. To use SCP, connect as the `root` user, not `admin`. If a custom user has the `User - System - Copy files` permission, or all access, then they may also utilize SCP.

Tip: SSH clients must be kept up-to-date. As time goes on, security standards evolve and the SSH server settings utilized by SSH servers will change. Outdated clients may not be able to connect using the strong security keys and algorithms required by `sshd`. If a client will not connect, check for an update from the vendor.

Enable Secure Shell

To enable the SSH daemon, check **Enable Secure Shell**. After saving with this option enabled, the firewall will generate SSH keys if they are not already present and then start the SSH daemon.

SSHd Key Only

This option controls which authentication methods the SSH daemon allows for clients. It can be set to one of the following values:

Password or Public Key Allows a user to authenticate with **either** a valid password or valid key. This is the default behavior.

Public Key Only Restricts authentication to only valid keys, passwords are not allowed.

Require Both Password and Public Key Requires a valid password **and** a valid key.

Key-based logins are a much more secure practice, though it does take more preparation to configure.

Add user keys for key-based login by editing users in the **User Manager** (*User Management and Authentication*). When editing a user, paste the allowed public keys into the **Authorized Keys** text field for the account.

Allow Agent Forwarding

Controls whether or not the SSH daemon allows agent forwarding for clients.

Agent forwarding allows a user to run an SSH agent on their client system and connect to the firewall, and then to other remote SSH servers using the key from their agent. In this case, the user does not need to have their private keys on the firewall but can still use key-based authentication to remote servers.

Use of an SSH agent can be considered a security issue in certain cases. Additionally, the firewall is not intended to be a general purpose SSH client or intermediate system, thus this feature is disabled by default.

SSH Port

Controls the port used by the SSH daemon to accept client connections. To change the port, type the new port into the **SSH Port** box.

Moving the SSH server to an alternate port provides a negligible security improvement, and frees up the port for other uses.

Tip: Brute force SSH scanners focus on hitting TCP port `22` but if the daemon is open to the Internet on another port, it will eventually be found and hit by scanners.

Best Practices for SSH

If this firewall is installed in an environment that requires leaving SSH access unrestricted by firewall rules, which is dangerous, we strongly recommended the following actions:

Change the SSH Port Moving to a random alternate port prevents log noise from many, but not all, brute-force SSH login attempts and casual scans. It can still be found with a port scan, however.

Force Key-Based Authentication Key-based authentication must always be used by publicly accessible SSH servers to eliminate the possibility of successful brute force attacks. Set **SSHd Key Only** to either *Public Key Only* or *Require Both Password and Public Key*.

Multiple unsuccessful logins from the same IP address will result in locking out the IP address trying to authenticate, but that alone is not considered sufficient protection.

Login Protection

The `sshguard` daemon is used by the firewall to protect against brute force logins for both the GUI and SSH connections. The options in this section fine-tune the behavior of this protection.

Threshold The total score value above which `sshguard` will block clients. Most attacks have a score of `10`, the default threshold value is `30`.

Blocktime The initial minimum number of seconds to block attackers who have exceeded the **Threshold** value. The default value is `120` seconds. Repeat offenders are blocked for increasingly longer amounts of time (1.5x for each repetition).

Note: Attackers are unblocked at random intervals so actual block time will be longer than stated. This prevents clients from predicting the timing to optimize targeted attacks.

Detection Time The amount of time, in seconds, attackers are remembered by `sshguard` since their last offense before it resets their score. Default is `1800` seconds.

Whitelist A list of subnets which are excluded from login protection. This lowers security but is generally acceptable from specific secure management networks.

For example, it may be necessary to add entries for network monitoring systems which probe the SSH port but do not login. Otherwise such systems may be flagged as attackers.

Serial Communications

If the firewall is running on hardware without a monitor or if it will be running "headless" (without keyboard and video attached), then the serial console can be enabled to maintain physical control, so long as the hardware has a serial port (not USB).

If hardware is detected which has no VGA port, the serial console is forced on and cannot be disabled, and the serial options are all hidden except for the speed.

Serial Terminal

When **Serial Terminal** is set, the operating system enables the console on the first serial port. This console will receive kernel boot messages and a menu after the firewall has finished booting. This will not disable the onboard keyboard and video console.

To connect to the serial console, use a null modem cable connected to a serial port or adapter on another PC or serial device.

See also:

For more information on connecting to a serial console, see *Connecting to a Serial Console* and *Start a Serial Client*.

When making any changes to the serial console, the firewall must be rebooted before they take effect.

Serial Console Speed

The default serial console speed is *115200* bps and almost all hardware works well at that speed. In rare cases, a slower speed may be required which can be set here by picking the desired speed from the **Serial Speed** drop-down.

When upgrading from an older version, this may remain at an older value such as *9600* or *38400* to match the BIOS on older hardware. Increasing the speed to *115200* is almost always safe and more useful than slower speeds.

Primary Console

On hardware with both the serial console enabled and a VGA port available, the **Primary Console** selector chooses which is the preferred console, so it will receive the boot log messages. Other OS kernel messages will show up on all console connections, and both consoles will have a usable menu.

In cases where the boot cannot complete, the preferred console must be used to resolve the problem, such as reassigning interfaces.

Console Menu

Normally the firewall always presents the menu on the console, and the menu will be available as long as someone has physical access to the console. In high-security environments this is not desirable.

This option adds password protection to the console. The console accepts the same usernames and passwords used to access the GUI. After setting this option, the firewall must be rebooted before it takes effect.

Note: While this will stop accidental key presses and keep out casual users, this is by no means a perfect security method. A knowledgeable person with physical access can still reset the passwords (see *Forgotten Password with a Locked Console*). Consider other physical security methods if console security is a requirement.

8.6.2 Firewall/NAT Tab

Firewall Advanced

IP Do-Not-Fragment compatibility

This option is a workaround for operating systems which generate fragmented packets with the "don't fragment" (DF) bit set. Linux NFS (Network File System) is known to do this, as well as some VoIP implementations.

When this option is enabled, the firewall will not drop these malformed packets but instead it will clear the DF bit. The firewall will also randomize the IP identification field of outgoing packets to compensate for operating systems that set the DF bit but set a zero IP identification header field.

IP Random ID generation

If **Insert a stronger ID into IP header of packets passing through the filter** is checked, the firewall replaces the IP identification field of packets with random values to compensate for operating systems that use predictable values. This option only applies to packets that are not fragmented after the optional packet reassembly.

Firewall Optimization Options

The optimization mode controls how the firewall expires state table entries:

Normal The standard optimization algorithm, which is optimal for most environments.

High Latency Used for high latency links, such as satellite links. Expires idle connections later than default.

Aggressive Expires idle connections quicker. More efficient use of CPU and memory but can drop legitimate connections earlier than expected. This option can also improve performance in high traffic deployments with lots of connections, such as web services.

Conservative Tries to avoid dropping any legitimate connections at the expense of increased memory usage and CPU utilization. Can aid in environments that require long-lived but mostly idle UDP connections, such as VoIP.

The table *Firewall Optimization Details* contains the values chosen by PF for each optimization algorithm. The values are taken from the PF source code. The first line is the raw value, second line is human readable:

Table 1: Firewall Optimization Details

	Normal	High Latency	Conservative	Aggressive
tcp.first First TCP packet	60 1min	180 3min	3600 60min	30 30sec
tcp.opening No response yet	30 30sec	35 35sec	900 15min	5 5sec
tcp.established Established	86400 24h	86400 24h	432000 5days	18000 5h
tcp.closing Half closed	900 15min	905 15min + 5sec	3600 1h	60 60sec
tcp.finwait Got both FINs	45 45sec	50 50sec	600 10min	30 30sec
tcp.closed Got an RST	90 90sec	95 95sec	180 3min	30 30sec
tcp.tsdiff Allowed TS diff	30 30sec	60 60sec	60 60sec	10 10sec

Disable Firewall

When **Disable all packet filtering** is set, the firewall becomes a routing-only platform. This is accomplished by disabling `pf` entirely, and as a consequence, NAT is disabled since it is also handled by `pf`.

Tip: To disable *only* NAT, do not use this option. Consult *Disabling Outbound NAT* for more information on controlling outbound NAT behavior.

Disable Firewall Scrub

When set, the scrubbing option in `pf` is disabled. The `scrub` action in `pf` can interfere with NFS, and in rare cases, with VoIP traffic as well. By default, the firewall uses the `fragment reassemble` option which reassembles fragmented packets before sending them on to their destination, when possible. More information on the `scrub` feature of `pf` can be found in the OpenBSD PF Scrub Documentation.

Note: Disabling `scrub` also disables other features that rely on `scrub` to function, such as DF bit clearing and ID randomization. Disabling `scrub` does not disable MSS clamping if it is active for VPNs, or when an MSS value is configured on an interface.

Firewall Adaptive Timeouts

Adaptive Timeouts control state handling in `pf` when the state table is nearly full. Using these timeouts, a firewall administrator can control how states are expired or purged when there is little or no space remaining to store new connection states.

Adaptive Timeouts are enabled by default and the default values are calculated automatically based on the configured **Firewall Maximum States** value.

Adaptive Start Adaptive scaling is started once the state table reaches this level, expressed as a number of states. **Adaptive Start** defaults to 60% of **Firewall Maximum States**.

Adaptive End When the size of the state table reaches this value, expressed as a number of state table entries, all timeout values are assumed to be **zero**, which causes pf to purge all state entries immediately. This setting defines the scale factor, it should be set greater than the total number of states allowed. **Adaptive End** defaults to 120% of **Firewall Maximum States**.

When the number of connection states exceeds the threshold set by **Adaptive Start**, timeout values are scaled linearly with factor based on the number of states used between the Start and End state counts. The timeout adjustment factor is calculated as follows: (Number of states until the **Adaptive End** value is reached) / (Difference between the **Adaptive End** and **Adaptive Start** values).

Note: As an example, consider a firewall with **Adaptive Start** set to `600000`, **Adaptive End** set to `1200000` and **Firewall Maximum States** set to `1000000`. In this situation, when the state table size reaches `900000` entries the state timeouts will be scaled to 50% of their normal values.

(1,200,000 - 900,000) / (1,200,000 - 600,000) = 300,000 / 600,000 = 0.50, 50%

Continuing the example, when the state table is full at 1,000,000 states the timeout values will be reduced to 1/3 of their original values.

Tip: The state table usage indicator on the dashboard will change color and text when the state table size crosses these thresholds.

Firewall Maximum States

This value is the maximum number of connections the firewall can hold in its state table. The default size is calculated based on 10% of total RAM. This default value is sufficient for most installations, but can be adjusted higher or lower depending on the load and available memory.

Each state consumes approximately 1 KB of RAM, or roughly 1 MB of RAM for every 1000 states. The firewall must have adequate free RAM to contain the entire state table before increasing this value. Firewall states are discussed further in *Stateful Filtering*.

Tip: On a firewall with 8GB of RAM the state table would have a default size of approximately 800,000 states. A custom **Firewall Maximum States** value of 4,000,000 would consume about 4GB of RAM, half the available 8GB total.

Firewall Maximum Table Entries

This value defines the maximum number of entries that can exist inside of address tables used by the firewall for collections of addresses such as aliases, ssh/GUI lockout records, hosts blocked by snort alerts, and so on. By default this is 200,000 entries. If the firewall has features enabled which can load large blocks of address space into aliases such as URL Table aliases or the pfBlockerNG package, then increase this value to comfortably include at least double the total amount of entries contained in all aliases combined.

Firewall Maximum Fragment Entries

When scrub is enabled the firewall maintains a table of packet fragments waiting to be reassembled. By default this table can hold `5000` fragments. In rare cases a network may have an unusually high rate of fragmented packets which can require more space in this table. When this limit is reached, the following log message will appear in the main system log:

```
kernel: [zone: pf frag entries] PF frag entries limit reached
```

Static Route Filtering

The **Bypass firewall rules for traffic on the same interface** option applies if the firewall has one or more static routes defined. If this option is enabled, traffic that enters and leaves through the same interface will not be checked by the firewall. This may be required in situations where multiple subnets are connected to the same interface, to avoid blocking traffic that is passed through the firewall in one direction only due to asymmetric routing. See *Bypass Firewall Rules for Traffic on Same Interface* for a more in-depth discussion on that topic.

Disable Auto-added VPN rules

By default, when IPsec is enabled firewall rules are automatically added to the appropriate interface which will allow the tunnel to establish. When **Disable Auto-added VPN rules** is checked, the firewall will not automatically add these rules. By disabling these automatic rules, the firewall administrator has control over which addresses are allowed to connect to a VPN. Further information on these rules can be found at *VPNs and Firewall Rules*.

Disable Reply-To

In a Multi-WAN configuration the firewall has a beneficial default behavior that ensures traffic leaves the same interface it arrived through. This is accomplished using the `pf` keyword `reply-to` which is added automatically to interface tab firewall rules for WAN-type interfaces. When a connection matches a rule with `reply-to`, the firewall remembers the path through which the connection was made and routes the reply traffic back to the gateway for that interface.

Tip: WAN-type interfaces are interfaces which have a gateway set on their **Interfaces** menu entry configuration, or interfaces which have a dynamic gateway such as DHCP, PPPoE, or assigned OpenVPN, GIF, or GRE interfaces.

In situations such as bridging, this behavior is undesirable if the WAN gateway IP address is different from the gateway IP address of the hosts behind the bridged interface. Disabling `reply-to` will allow clients to communicate with the proper gateway.

Another case that has issues with `reply-to` involves static routing to other systems in a larger WAN subnet. Disabling `reply-to` in this case would help ensure that replies return to the proper router instead of being routed back to the gateway.

This behavior can also be disabled on individual firewall rules rather than globally using this option.

Disable Negate rules

In a Multi-WAN configuration traffic for directly connected networks and VPN networks typically must still flow properly when using policy routing. The firewall will insert rules to pass this local and VPN traffic without a gateway specified, to maintain connectivity. In some cases these negation rules can over-match traffic and allow more than intended.

Tip: The best practice is to create manual negation rules at the top of internal interfaces such as LAN. These rules should pass to local and VPN destinations without a gateway set on the rule, to honor the system routing table. These rules do not have to be at the top of the interface rules, but they must be above rules that have a gateway set.

Allow APIPA

Automatic Private IP Addressing (APIPA), or IPv4 Link-Local addressing, uses a special subnet of `169.254.0.0/16`. This traffic is for local links only (same L2), it must not be routed or traverse a firewall. As such, inbound traffic from these addresses is automatically blocked by internal firewall rules by default.

When **Allow APIPA traffic** is checked, the default block rules are removed, and user firewall rules can control the traffic.

There are some use cases which utilize these addresses for private communication on an interface, such as AWS VPC BGP, and in those cases, the option can be enabled along with carefully crafted manual firewall rules.

Warning: When this option is enabled, take care to never allow APIPA traffic to match policy routing rules. If APIPA traffic matches policy routing rules, behavior is unpredictable. There have been reports of such errors leading to packet loops and unexpectedly high resource usage. See Redmine Issue #2073 for more.

Aliases Hostnames Resolve Interval

This option controls how often hostnames in aliases are resolved and updated by the `filterdns` daemon. By default this is `300` seconds (5 minutes). In configurations with a small number of hostnames or a fast/low-load DNS server, decrease this value to pick up changes faster.

Check Certificate of Alias URLs

When **Verify HTTPS certificates when downloading alias URLs** is set, the firewall will require a valid HTTPS certificate for web servers used in URL table aliases. This behavior is more secure, but if the web server is private and uses a self-signed certificate, it can be more convenient to ignore the validity of the certificate and allow the data to be downloaded.

Warning: The best practice is to always use a server certificate with a valid chain of trust for this type of role, rather than weakening security by allowing a self-signed certificate.

Bogon Networks

The **Update Frequency** drop-down for **Bogon Networks** controls how often these lists are updated. Further information on bogon networks may be found in *Block Bogon Networks*.

Network Address Translation

NAT Reflection for Port Forwards

The **NAT Reflection mode for port forwards** option controls how NAT reflection is handled by the firewall. These NAT redirect rules allow clients to access port forwards using the public IP addresses on the firewall from within local internal networks.

See also:

Refer to *NAT Reflection* for a discussion on the merits of NAT Reflection when compared to other techniques such as Split DNS.

There are three possible modes for NAT Reflection:

Disabled The default value. When disabled, port forwards are only accessible from WAN and not from inside local networks.

Pure NAT This mode uses a set of NAT rules to direct packets to the target of the port forward. It has better scalability, but it must be possible to accurately determine the interface and gateway IP address used for communication with the target at the time the rules are loaded. There are no inherent limits to the number of ports other than the limits of the protocols. All protocols available for port forwards are supported.

When this option is enabled, **Automatic Outbound NAT for Reflection** must also be enabled if the clients and servers are in the same local network.

NAT + Proxy *NAT + proxy* mode uses a helper program to send packets to the target of the port forward. The connection is received by the reflection daemon and it acts as a proxy, creating a new connection to the local server. This behavior puts a larger burden on the firewall, but is useful in setups where the interface and/or gateway IP address used for communication with the target cannot be accurately determined at the time the rules are loaded. *NAT + Proxy* reflection rules are not created for ranges larger than 500 ports and will not be used for more than 1000 ports total between all port forwards. This feature only supports TCP port forwards.

Individual NAT rules have the option to override the global NAT reflection configuration, so they may have NAT reflection forced on or off on a case-by-case basis.

Reflection Timeout

The **Reflection Timeout** setting forces a timeout on connections made when performing NAT reflection for port forwards in *NAT + Proxy* mode. If connections are staying open and consuming resources, this option can mitigate that issue.

NAT Reflection for 1:1 NAT

When checked, this option adds additional reflection rules which enable access to 1:1 mappings of external IP addresses from internal networks. This gives the same functionality that already exists for port forwards, but for 1:1 NAT. There are complex routing scenarios that may render this option ineffective.

This option only affects the *inbound* path for 1:1 NAT, not outbound. The underlying rule style is similar to the *Pure NAT* mode for port forwards. As with port forwards, there are per-entry options to override this behavior.

Automatic Outbound NAT for Reflection

When checked, this option automatically creates outbound NAT rules which assist reflection rules that direct traffic back out to the same subnet from which it originated. These additional rules allow Pure NAT and 1:1 NAT Reflection to function fully when the clients and servers are in the same subnet. In most cases, this box must be checked for NAT Reflection to work.

Note: This behavior is necessary because when clients and servers are in the same subnet, the traffic source must be changed so that the connection appears to originate from the firewall. Otherwise, the return traffic will bypass the firewall and the connection will not succeed.

TFTP Proxy

The built-in TFTP proxy will proxy connections to TFTP servers outside the firewall, so that client connections may be made to remote TFTP servers. Ctrl-click or shift-click to select multiple entries from the list. If no interfaces are chosen, the TFTP proxy service is deactivated.

State Timeouts

The **State Timeout** section allows fine-tuning of the state timeouts for various protocols. These are typically handled automatically by the firewall and the values are dictated by the *Firewall Optimization Options* options. In rare cases, these timeouts may need adjusted up or down to account for irregularities in device behavior or site-specific needs.

All of the values are expressed in *seconds*, and control how long a connection in that state will be retained in the state table.

See also:

Descriptions in the following options reference firewall state conditions as described in *Interpreting States*.

TCP First The first packet of a TCP connection.

TCP Opening The state before the destination host has replied (e.g. `SYN_SENT:CLOSED`).

TCP Established An established TCP connection where the three-way handshake has been completed.

TCP Closing One side has sent a TCP FIN packet.

TCP FIN Wait Both sides have exchanged FIN packets and the connection is shutting down. Some servers may continue to send packets during this time.

TCP Closed One side has sent a connection reset (TCP RST) packet.

TCP Tsdiff The allowed TCP timestamp difference.

UDP First The first UDP packet of a connection has been received.

UDP Single The source host has sent a single packet but the destination has not replied (e.g. `SINGLE:NO_TRAFFIC`).

UDP Multiple Both sides have sent packets.

ICMP First An ICMP packet has been received.

ICMP Error An ICMP error was received in response to an ICMP packet.

Other First, Other Single, Other Multiple The same as **UDP**, but for other protocols.

8.6.3 Networking Tab

IPv6 Options

Allow IPv6

The **Allow IPv6** option controls a set of block rules which prevent IPv6 traffic from being handled by the firewall.

Note: This option does not disable IPv6 functions or prevent it from being configured, it only controls traffic flow.

When the option is enabled, IPv6 traffic will be allowed when permitted by firewall rules and/or automatic rules, depending on the firewall configuration. This option is enabled by default on new configurations.

When the option is unchecked, all IPv6 traffic will be blocked. This behavior is similar to how IPv6 was treated before it was supported by pfSense® software. Configurations imported from or upgraded from versions older than 2.1 will have this option unchecked, so they behave consistently after upgrade.

IPv6 over IPv4 Tunneling

The **Enable IPv6 over IPv4 Tunneling** option enables forwarding for IP protocol 41/RFC 2893 to an IPv4 address specified in the **IPv4 address of Tunnel Peer** field.

When configured, this forwards all incoming protocol 41/IPv6 traffic to a host behind this firewall instead of handling it locally.

Tip: Enabling this option does not add firewall rules to allow the protocol 41 traffic. A rule must exist on the WAN interface to allow the traffic to pass through to the local receiving host.

Prefer IPv4 over IPv6

When set, this option causes the firewall *itself* to prefer sending traffic to IPv4 hosts instead of IPv6 hosts when a DNS query returns results for both.

In rare cases when the firewall has partially configured, but not fully routed, IPv6 this can allow the firewall to continue reaching Internet hosts over IPv4.

Note: This option controls the behavior of the firewall itself, such as when polling for updates, package installations, downloading rules, and fetching other data. It cannot influence the behavior of clients behind the firewall.

IPv6 DNS Entry

This option controls whether or not the firewall creates local DNS entries for the firewall itself with IPv6 addresses, when available.

By default (unchecked), the firewall automatically adds DNS entries for itself using its local IPv4 and IPv6 interface addresses. In some cases, such as with dynamic IPv6 addresses like tracked interfaces, the IPv6 address may disappear or change and clients may attempt to use an outdated address until their cached DNS response expires.

When the option is checked, the firewall only adds DNS entries for its IPv4 addresses.

DHCP6 DUID

This option controls the DHCPv6 Unique Identifier (DUID) used by the firewall when requesting an IPv6 address. The firewall generates a DUID automatically, but in some cases, an administrator may want to use a different DUID. For example, if the operating system was reinstalled and the firewall should use the same DUID it had in the past, or if an upstream network administrator requires a specific DUID.

Note: Most users do not need to change this to any specific value, the default behavior is fine for nearly all environments. When in doubt, leave it alone unless directed to change it by an upstream network provider.

There are several possible DUID formats that this option can accept, chosen by the drop-down menu. When a format is chosen, the GUI displays a different set of input boxes specific to the selected format. The exact format depends upon the needs of the network administrator (e.g. ISP, datacenter, etc) and they would provide the format and values.

The available DUID formats are:

Raw DUID DUID represented exactly as observed in a DUID file or in logs. Entered as:

Raw DUID A single text area in which the DUID can be entered.

This option also includes a  **Copy DUID** button which copies the DUID from the placeholder (automatically generated by the firewall) into the text box so that the existing DUID can easily be placed into the configuration.

DUID-LLT DUID format with Link-Layer Address Plus Time. Entered as:

Time Time (in seconds) since January 1st, 2000 UTC

Link-Layer Address The link-layer address (MAC) of an interface on the firewall in the format `xx:xx:xx:xx:xx:xx`.

DUID-EN DUID assigned by a vendor based on Enterprise Number. Entered as:

Enterprise Number IANA Private Enterprise Number of the vendor.

Identifier Variable length identifier in the format `xx:xx:xx:xx`. The length depends upon the vendor.

DUID-LL DUID based on only Link-Layer Address. Entered as:

Link-Layer Address The link-layer address (MAC) of an interface on the firewall in the format `xx:xx:xx:xx:xx:xx`.

DUID-UUID DUID based on the host Universally Unique Identifier (UUID). Entered as:

DUID-UUID The UUID for this host in the format `nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn`

Network Interfaces

Hardware Checksum Offloading

When checked, this option disables hardware checksum offloading on the network cards. Checksum offloading is usually beneficial as it allows the checksum to be calculated (outgoing) or verified (incoming) in hardware at a much faster rate than it could be handled in software.

Note: When checksum offloading is enabled, a packet capture will see empty (all zero) or flag incorrect packet checksums. These are normal when checksum handling is happening in hardware.

Checksum offloading is broken in some hardware, particularly Realtek cards and virtualized/emulated cards such as those on Xen/KVM. Typical symptoms of broken checksum offloading include corrupted packets and poor throughput performance.

Tip: In virtualization cases such as Xen/KVM it may be necessary to disable checksum offloading on the host as well as the VM. If performance is still poor or has errors on these types of VMs, switch the type of NIC if possible.

Hardware TCP Segmentation Offloading

Checking this option will disable hardware TCP segmentation offloading (TSO, TSO4, TSO6). TSO causes the NIC to handle splitting up packets into MTU-sized chunks rather than handling that at the OS level. This can be faster for servers and appliances as it allows the OS to offload that task to dedicated hardware, but when acting as a firewall or router this behavior is **highly undesirable** as it actually increases the load as this task has already been performed elsewhere on the network, thus breaking the end-to-end principle by modifying packets that did not originate on this host.

> **Warning:** This option is not desirable for routers and firewalls, but can benefit workstations and appliances. It is disabled by default, and should remain disabled unless the firewall is acting primarily or solely in an appliance/endpoint role.
>
> Do not uncheck this option unless directed to do so by a support representative. This offloading is broken in some hardware drivers, and can negatively impact performance on affected network cards and roles.

Hardware Large Receive Offloading

Checking this option will disable hardware large receive offloading (LRO). LRO is similar to TSO, but for the incoming path rather than outgoing. It allows the NIC to receive a large number of smaller packets before passing them up to the operating system as a larger chunk. This can be faster for servers and appliances as it offloads what would normally be a processing-heavy task to the network card. When acting as a firewall or router this is **highly undesirable** as it delays the reception and forwarding of packets that are not destined for this host, and they will have to be split back up again on the outbound path, increasing the workload significantly and breaking the end-to-end principle.

> **Warning:** This option is not desirable for routers and firewalls, but can benefit workstations and appliances. It is disabled by default, and should remain disabled unless the firewall is acting primarily or solely in an appliance/endpoint role.
>
> Do not uncheck this option unless directed to do so by a support representative. This offloading is broken in some hardware drivers, and can negatively impact performance on affected network cards and roles.

hn ALTQ Support

Checking this option will enable support for ALTQ traffic shaping on `hn(4)` network interfaces in Hyper-V.

For ALTQ to work on `hn(4)` interfaces, the operating system must disable the multi-queue API which may reduce the system capability to handle traffic. The administrator must decide if this reduction in performance is worth the benefit of traffic shaping.

The firewall must be rebooted for this setting to take effect.

Suppress ARP messages

The firewall makes a log entry in the main system log when an IP address appears to switch to a different MAC address. This log entry notes that the device has moved addresses, and records the IP address and the old and new MAC addresses.

This event can be completely benign behavior (e.g. NIC teaming on a Microsoft server, a device being replaced) or a legitimate client problem (e.g. IP conflict), and it could show up constantly or rarely if ever. It all depends on the network environment.

The best practice is to allow these ARP messages to be printed to log since there is a chance it will report a problem worth the attention of a network administrator. However, if the network environment contains systems which generate these messages while operating normally, suppressing the errors can make the system log more useful as it will not be cluttered with unneeded log messages.

Reset All States

When set, if an interface IP address changes, the firewall will reset the entire state table instead of only clearing states for the old interface IP address.

This behavior is potentially disruptive, and is off by default. In single WAN environments, this is not typically any more disruptive than the WAN address changing, since clients already have to reestablish all connections.

In most cases, this behavior is not necessary, but it can help in certain situations where WAN addresses change rapidly and the normal behavior misses states for former IP addresses.

8.6.4 Miscellaneous Tab

Proxy Support

If this firewall resides in a network which requires a proxy for outbound Internet access, enter the proxy options in this section so that requests from the firewall for items such as packages and updates will be sent through the proxy.

Proxy URL This option specifies the location of the proxy for making outside connections. It must be an IP address or a fully qualified domain name.

Proxy Port The port to use when connecting to the proxy URL. By default the port is `8080` for HTTP proxy URLs, and `443` for SSL proxy URLs. The port is determined by the proxy, and may be a different value entirely (e.g. `3128`). Check with the proxy administrator to find the proper port value.

Proxy Username If required, this is the username that is sent for proxy authentication.

Proxy Password If required, this is the password associated with the username set in the previous option.

Load Balancing

When pfSense® software is directed to perform load balancing, successive connections will be redirected in a round-robin manner to a gateway, balancing the load across all available paths. The options in this section alter or fine-tune that behavior.

Sticky Connections When active, connections from the same source are sent through the same gateway, rather than being sent in a purely round-robin manner.

This "sticky" association will exist as long as states are in the table for connections from a given source address (e.g. the IP address of a user). Once the states for that source expire, so will the

sticky association. Further connections from that source host will be redirected to the next available gateway in the group.

This behavior can help with protocols such as HTTPS and FTP, where the server may be strict about all connections coming from the same source. The downside of this behavior is that balancing is not as efficient, a heavy user could dominate a single WAN rather than having their connections spread out.

Source Tracking Timeout Controls how long the sticky association will be maintained for a host after the all of the states from that host expire. The value is specified in seconds.

By default, this value is not set, so the association is removed as soon as the states expire. If sticky connections appear to work initially but seem to stop partway through sessions, increase this value to hold an association longer. Web browsers often hold open connections for a while as users are on a site, but if there is a lot of idle time, connections may be closed and states may expire.

Power Savings

When **Enable PowerD** is checked, the `powerd` daemon is started. This daemon monitors the system and can lower or raise the CPU frequency based on system activity. If processes need the power, the CPU speed will be increased as needed. This option will lower the amount of heat a CPU generates, and may also lower power consumption.

Note: The behavior of this option depends greatly on the hardware in use. In some cases, the CPU frequency may lower but have no measurable effect on power consumption and/or heat, where others will cool down and use considerably less power. It is considered safe to run, but is left off by default unless supported hardware is detected.

The mode for `powerd` may also be selected for three system states:

AC Power Normal operation connected to AC power.

Battery Power Mode to use when the firewall is running on battery. Support for battery power detection varies by hardware.

Unknown Power Mode used when `powerd` cannot determine the power source.

Four modes choices exist for each of these states:

Maximum Keeps the performance as high as possible at all times.

Minimum Keeps performance at its lowest, to reduce power consumption.

Adaptive Tries to balance savings by decreasing performance when the system is idle and increasing when busy.

Hiadaptive Similar to adaptive but tuned to keep performance high at the cost of increased power consumption. It raises the CPU frequency faster and drops it slower. This is the default mode.

Note: Some hardware requires `powerd` running to operate at its maximum attainable CPU frequency. If the firewall device does not have `powerd` enabled but always runs at what appears to be a low CPU frequency, enable `powerd` and set it to *Maximum* for at least the **AC Power** state.

Watchdog

Certain firewall hardware includes a **Watchdog** feature which can reset the hardware when the watchdog daemon can no longer interface with the hardware after a specified timeout. This can increase reliability by resetting a unit when a hard lock is encountered that might otherwise require manual intervention.

The downside to any hardware watchdog is that any sufficiently busy system may be indistinguishable from one that has suffered a hard lock.

Enable Watchdog When checked, the `watchdogd` daemon is run which attempts to latch onto a supported hardware watchdog device.

Watchdog Timeout The time, in seconds, after which the device will be reset if it fails to respond to a watchdog request. If a firewall regularly has a high load and triggers the watchdog accidentally, increase the timeout.

Cryptographic & Thermal Hardware

Cryptographic Hardware

There are a few options available for accelerating cryptographic operations via hardware. Some are built into the kernel, and others are loadable modules.

The following choices are available, depending on hardware:

BSD Crypto Device Loads the BSD Crypto device module (`cryptodev`) so it can be used by other available acceleration devices. Most accelerator drivers hook into the `crypto(9)` framework in FreeBSD, so many aspects of the system will automatically use acceleration for supported ciphers when this module is loaded.

AES-NI CPU-based Acceleration Loads the **AES-NI** (Advanced Encryption Standard, New Instructions) kernel module. Notably, the `aesni` module will accelerate operations for AES-GCM, available in IPsec.

Support for AES-NI is built into many recent Intel and some AMD CPUs. Check with the OEM for specific CPU or SoC support.

Speeds with AES-NI vary by support of the underlying software. IPsec speed will be greatly increased with AES-NI loaded provided that AES-GCM is used and properly configured.

AES-NI and BSD Crypto Device Loads both the AES-NI and BSD Crypto Device modules together, which is the optimal configuration in most cases. Choose this unless a specific environment or configuration is found to work better without it.

SafeXcel and BSD Crypto Device Loads both the `safexcel` and the BSD Crypto Device modules. SafeXcel acceleration hardware is found on some ARM systems sold by Netgate, such as the SG-3100.

There are other supported cryptographic devices with drivers built into the kernel. One example is the driver for the Marvell Cryptographic Engine and Security Accelerator (CESA) chipset, which is found on some ARM systems sold by Netgate, such as the SG-1100 and SG-2100.

In most cases, if a supported accelerator chip is detected by the firewall, it will be shown in the **System Information** widget on the dashboard or in the system log at boot time.

Note: Certain special cases also exist where software can detect and use acceleration hardware directly, even without drivers loaded. One example is OpenSSL, which directly supports AES-NI. Thus, even without the driver loaded,

software which utilizes encryption through OpenSSL can still take advantage of AES-NI acceleration.

Thermal Sensors

The firewall can read temperature data from a few sources to display on the dashboard. If the firewall has a supported CPU, selecting a thermal sensor will load the appropriate driver to read its temperature.

Note: Temperature data can be displayed by the **Thermal Sensors** *dashboard widget* or via `sysctl`.

The following sensor types are supported:

None/ACPI The firewall will attempt to read the temperature from an ACPI-compliant motherboard sensor if one is present, otherwise no sensor readings are available.

Intel Core Loads the `coretemp` module which supports reading thermal data from Intel core-series CPUs and other modern Intel CPUs using their on-die sensors, including Atom-based processors.

AMD K8, K10, and K11 Loads the `amdtemp` module which supports reading thermal data from modern AMD CPUs using their on-die sensors.

If the firewall does not have a supported thermal sensor chip, this option will have no effect. To unload the selected module, set this option to *None/ACPI* and then reboot.

Note: The `coretemp` and `amdtemp` modules report thermal data directly from the CPU core. This may or may not be indicative of the temperature elsewhere in the system. Case temperatures can vary greatly from temperatures on the CPU die.

Kernel Page Table Isolation (PTI)

Kernel PTI is a method for working around CPU vulnerabilities such as Meltdown. By exploiting that vulnerability without Kernel PTI, kernel memory could be accessed by unprivileged users on affected CPUs.

Note: While more secure, this protection can incur a performance penalty. If untrusted users do not have access to run arbitrary code on the firewall, it can be disabled without significant security risk.

Kernel PTI is active by default only on CPUs affected by the vulnerability.

This option forces the workaround off, and requires a reboot to change.

If a vulnerable CPU is not detected, PTI is disabled by default and this option will have no effect.

The current state of Kernel PTI is printed below the option.

Microarchitectural Data Sampling (MDS) Mitigation

Microarchitectural Data Sampling (MDS) mitigation is a method for working around weaknesses in Intel CPUs which support hyperthreading. By exploiting MDS without mitigation in place, kernel memory could be accessed by unprivileged users on affected CPUs.

Note: While more secure, this protection can incur a performance penalty. If untrusted users do not have access to run arbitrary code on the firewall, it can be disabled without significant security risk.

This option controls which method of MDS mitigation is used, if any. Changing the option requires a reboot to activate. The following modes are available:

Default The default operating system behavior. As of this writing, the default behavior is to disable MDS mitigation.

Mitigation Disabled Forcefully disable MDS mitigation.

VERW instruction (microcode) mitigation enabled Use VERW instruction mitigation, implemented in CPU microcode, to mitigate MDS. This is the fastest and most optimal way to mitigate MDS, but it requires support in the CPU microcode for this instruction.

Software sequence mitigation enabled Mitigates MDS by using software sequences, which is much slower, but safer.

Automatic VERW or Software selection When set to Automatic, the operating system will attempt to use VERW instructions if they are available and software in all other cases.

The current state of MDS mitigation is printed below the option.

Schedules

The **Do not kill connections when schedule expires** option controls whether or not states are cleared when a scheduled rule transitions into a state that would block traffic. If unchecked, connections are terminated when the schedule time has expired. If checked, connections are left alone and will not be automatically closed by the firewall.

Gateway Monitoring

State Killing on Gateway Failure

When using Multi-WAN, by default the monitoring process will not flush states when a gateway goes into a down state. Flushing states for each gateway event can be disruptive in situations where a gateway is unstable.

The **Flush all states when a gateway goes down** option overrides the default behavior, clearing states for **all** existing connections when **any** gateway fails. Clearing states can help redirect traffic for long-lived connections such as VoIP phone/trunk registrations to another WAN, but it can also disrupt ongoing connections if a lesser-used gateway is flapping which would still kill all states when it fails.

More information on how this impacts Multi-WAN can be found in *State Killing/Forced Switch*.

Warning: When this is triggered, the entire state table is cleared. This is necessary because it is not possible to kill all states for the failing WAN and the LAN-side states associated with the failing WAN. Removing states on the WAN side alone is ineffective, the LAN-side states must be cleared as well.

Skip Rules When Gateway is Down

By default, when a rule has a specific gateway set and this gateway is down, the gateway is omitted from the rule and traffic is sent via the default gateway.

The **Do not create rules when gateway is down** option overrides that behavior and the entire rule is omitted from the ruleset when the gateway is down. Instead of flowing via the default gateway, the traffic will match a different rule instead. This is useful if traffic must only ever use one specific WAN and never flow over any other WAN.

Tip: When utilizing this option, create a reject or block rule underneath the policy routing rule with the same matching criteria. This will prevent the traffic from potentially matching other rules below it in the ruleset and taking an unintended path.

RAM Disk Settings

The `/tmp` and `/var` directories are used for writing files and holding data that is temporary and/or volatile. Using a RAM disk can reduce the amount of writing that happens on disks in the firewall. Modern SSDs do not have disk write concerns as older drives once did, but it can still be a concern when running from lower quality flash storage such as USB thumb drives.

This behavior has the benefit of keeping most of the writes off of the disk in the base system, but packages may yet write frequently to the drive. It also requires additional handling to ensure data such as RRD graphs and DHCP leases are retained across reboots. Data for both is saved during a proper shutdown or reboot, and also periodically if configured.

Use RAM Disks When checked, a memory disk is created at boot time for `/tmp` and `/var/` and the associated structure is initialized. When this setting is toggled, a reboot is required and forced on save.

> **Warning:** The size of RAM disks is limited by the amount of available kernel memory. The actual limit is calculated and printed in the GUI underneath the size options.

/tmp RAM Disk Size The size of the `/tmp` RAM disk, in MiB. The default value is `40`, but should be set higher if there is available RAM and kernel memory.

/var RAM Disk Size The size of the `/var` RAM disk, in MiB. The default value is `60`, but should be set much higher, especially if packages will be used. `512-1024` is a better starting point, depending on the available firewall RAM and kernel memory.

Periodic RAM Disk Data Backups These options control how frequently data in RAM disks is backed up. If the firewall is rebooted unexpectedly, the last backup is restored when the firewall boots. The lower the value, the less data that will be lost in such an event, but more frequent backups write more to the disk.

RRD Data The time, in hours, between periodic backups of RRD files containing graph data.

DHCP Leases The time, in hours, between periodic backups of the DHCP lease databases.

Log Directory The time, in hours, between periodic backups of the system log directory.

Warning: Aside from the points mentioned above, there are several items to be cautious about when choosing whether or not to use the RAM disk option. Used improperly, this option can lead to data loss or other unexpected failures.

Utilize remote syslog to send the logs to another device on the network rather than risking losing data from unexpected outages.

Packages may not properly account for the use of RAM disks, and may not function properly at boot time or in other ways. Test each package, including whether or not it works immediately after a reboot.

These are RAM disks, so the amount of RAM available to other programs will be reduced by the amount of space used by the RAM disks. For example if the firewall has 2GB of RAM, and has 512MB for `/var` and 512MB for `/tmp`, then only 1GB of RAM will be available to the OS for general use.

Special care must be taken when choosing a RAM disk size, which is discussed in the following section.

RAM Disk Sizes

Setting a size too small for `/tmp` and `/var` can backfire, especially when it comes to packages. The suggested sizes on the page are an **absolute minimum** and often much larger sizes are required. The most common failure is that when a package is installed, and parts of the package touch places in both `/tmp` and `/var` and it can ultimately fill up the RAM disk and cause other data to be lost. Another common failure is setting `/var` as a RAM disk and then forgetting to move a squid cache to a location outside of `/var` - if left unchecked, it will fill up the RAM disk.

For `/tmp`, a minimum of `40` MiB is required. For `/var` a minimum of `60` MiB is required. To determine the proper size, check the current usage of the `/tmp` and `/var` directories before making a switch. Check the usage several times over the course of a few days so it is not caught at a low point. Watching the usage during a package installation adds another useful data point.

Hard Disk Standby

The **Hard disk standby time** option activates power management for disk drives in the firewall. The drop-down field sets the number of minutes that the disk can be idle before going into standby mode.

Using standby mode is not necessary for SSD or flash media. For traditional spinning platter hard disks, it may result in power savings and can potentially lengthen the disk lifetime by saving wear, at a cost of slower disk access when resuming from an idle state. Actual results entirely depend on the hardware involved.

The default behavior is *Always On* which prevents the disk from entering standby mode.

Installation Feedback

When this option is set, the firewall will not send its Netgate Device ID when making requests to Netgate servers.

8.6.5 System Tunables Tab

The **System Tunables** tab under **System > Advanced** provides a means to set run-time FreeBSD system tunables, also known as sysctl OIDs.

Tip: In most cases, the best practice is to leave these tunables at their default values.

Firewall administrators familiar with FreeBSD, or users doing so under the direction of a developer or support representative, may want to adjust or add values on this page so that they will be set as the system starts.

Note: The tunables on this page are different from **Loader Tunables**. **Loader Tunables** are read-only values once the system has booted, and those values must be set in `/boot/loader.conf.local`.

Creating and Editing Tunables

To edit an existing tunable, click  .

To create a new tunable, click  **New** at the top of the list.

When editing or creating a tunable, the following fields are available:

Tunable The `sysctl` OID to set

Value The value to which the **Tunable** will be set.

> **Note:** Some values have formatting requirements. Due to the vast number of sysctl OIDs, the GUI does not validate that the given **Value** will work for the chosen **Tunable**.

Description An optional description for reference.

Click **Save** when the form is complete.

Tunable OIDs and Values

There are many OIDs available from `sysctl`, some of them can be set, some are read only outputs, and others must be set before the system boots as Loader Tunables. The full list of OIDs and their possible values is outside the scope of this documentation, but for those interested in digging a little deeper, The sysctl manual page from FreeBSD contains detailed instructions and information.

8.6.6 Notifications

The firewall can notify administrators of important events and errors by displaying an alert in the menu bar, indicated by the  icon.

In addition to GUI notifications, the firewall also supports the following remote notification methods:

- E-mail using SMTP
- Telegram notification API
- Pushover notification API

General Settings

Certificate Expiration When set, the firewall will issue notifications when certificates approach their expiration date, so that administrators can take corrective action to renew or replace them. Notifications are also sent for expired certificates.

The expiration times are checked daily, and notifications are displayed in the GUI and sent remotely.

Certificate Expiration Threshold The value, in days, at which certificates are considered to be approaching their expiration date.

The default value is currently `27` days. Certificates from Let's Encrypt (*ACME package*) typically renew when they have around 30 days before they expire. The default value is long enough that it does not notify unnecessarily, but with enough time left that problems can be corrected.

Tip: If certificates are imported into the firewall from third party sources which take longer to process, increase this value sufficiently to give administrators enough notice to obtain an updated replacement certificate before the expiration date.

SMTP E-mail

E-mail notifications are delivered by a direct SMTP connection to a mail server. The server must be configured to allow relaying from the firewall or accept authenticated SMTP connections.

Disable SMTP When checked, the firewall will not send SMTP notifications. This is useful to silence notifications while keeping SMTP settings in place for use by other purposes such as packages that utilize e-mail.

E-mail server The hostname or IP address of the e-mail server through which the firewall will send notifications.

SMTP Port of E-mail server The port to use when communicating with the SMTP server. The most common ports are `25` and `587`.

In many cases, `25` will not work unless it is to a local or internal mail server. Providers frequently block outbound connections to port `25`, so use `587` (the Submission port) when possible.

Connection Timeout to E-Mail Server The length of time, in seconds, that the firewall will wait for an SMTP connection to complete.

Secure SMTP Connection When set, the firewall will attempt an SSL/TLS connection when sending e-mail. The server must accept SSL/TLS connections or support STARTTLS.

Validate SSL/TLS When set, the certificate presented by the mail server is checked for validity against the root certificates trusted by the firewall. Ensuring this validity is the best practice.

In some rare cases a mail server may have a self-signed certificate or a certificate that otherwise fails validation. Unchecking this option will allow notifications to be sent to these servers using SSL/TLS. In this case, communication is still encrypted, but the identity of the server cannot be validated.

From e-mail address The e-mail address for the `From:` header in notification messages, which specifies the source. Some SMTP servers attempt to validate this address so the best practice is to use a real address in this field. This is commonly set to the same address as **Notification E-mail address**.

Notification E-mail address The e-mail address for the `To:` header of the message, which is the destination where the notification e-mails will be delivered by the firewall.

Notification E-Mail Auth Username Optional. If the mail server requires a username and password for authentication, enter the username here.

Notification E-Mail Auth Password Optional. If the mail server requires a username and password for authentication, enter the password here and in the confirmation field.

Notification E-mail Auth Mechanism This field specifies the authentication mechanism required by the mail server. The majority of e-mail servers work with *PLAIN* authentication, others such as MS Exchange may require *LOGIN* style authentication.

Click  **Save** at the bottom of the page to store the settings before proceeding.

Click  **Test SMTP Settings** to generate a test notification and send it via SMTP using the previously stored settings. Save settings before clicking this button.

Startup/Shutdown Sound

If the firewall hardware has a PC speaker, it will play a sound when startup finishes and again when a shutdown is initiated.

Check **Disable the startup/shutdown beep** to prevent the firewall from playing these sounds.

Telegram

The notification system supports the Telegram API which can send notifications to desktops and mobile devices, among others.

Note: Using the Telegram API requires a Telegram Bot and its associated API key.

Enable Telegram When set, the firewall will attempt to send remote notifications using the Telegram API and the settings in this section.

API Key Required. The Telegram Bot API key the firewall will use to authenticate with the Telegram API server.

Chat ID The destination for the notifications. This can be a chat ID number for private notifications, or a channel `@username` for public notifications.

Click  **Save** at the bottom of the page to store the settings before proceeding.

Click  **Test Telegram Settings** to generate a test notification and send it using the Telegram API with the previously stored settings. Save settings before clicking this button.

Pushover

The notification system supports the Pushover API which can send notifications to desktops and mobile devices, among others.

Note: Using the Pushover API requires a Pushover account user key and API key (Pushover Registration).

Enable Pushover When set, the firewall will attempt to send remote notifications using the Pushover API and the settings in this section.

API Key Required. The Pushover API Key (Pushover Registration) the firewall will use to authenticate with the Pushover API server.

User Key Required. The User Key (Pushover Registration) of the Pushover account to which the **API Key** belongs.

Notification Sound The notification sound that the end user device (Phone, etc) will play when notification messages are sent by the firewall.

See also:

For a list of sounds and audio, see the Pushover API Notification Sounds Documentation.

Message Priority The message priority for firewall notifications.

Note: For more information about the priorities and their meanings, see the Pushover API Priority Documentation.

The following priorities are available:

Normal Default setting. May trigger sound, vibration, and notification display depending on the user settings and client platform.

Lowest No sound or vibration, but increases the notification count on some platforms.

Low No sound or vibration. May trigger a notification display depending on the user settings and client platform.

High Always play sound and vibrate. Bypasses pre-set quiet hours. Notification display is highlighted in red.

Emergency Similar to *High* priority, but the notification is repeated until acknowledged by the user.

Emergency Priority Notification Retry Interval The amount of time, in seconds, the Pushover servers will send the same notification for *Emergency* priority notifications until the notification is acknowledged.

This parameter must have a value of at least `30` seconds between retries. Default is `60` seconds (1 minute).

Emergency Priority Notification Expiration The duration, in seconds, for which *Emergency* priority notifications will be retried until the notification is acknowledged. Notifications will be resent at intervals determined by the value of **Emergency Priority Notification Retry Interval**.

This parameter must have a maximum value of at most `10800` seconds (3 hours). Default is `300` seconds (5 minutes).

Click  **Save** at the bottom of the page to store the settings before proceeding.

Click  **Test Pushover Settings** to generate a test notification and send it using the Pushover API with the previously stored settings. Save settings before clicking this button.

8.7 Console Menu Basics

Basic configuration and maintenance tasks can be performed from the pfSense® system console. The console is available using a keyboard and monitor, serial console, or by using SSH. Access methods vary depending on hardware. Below is an example of what the console menu will look like, but it may vary slightly depending on the version and platform:

```
WAN (wan)       -> vmx0       -> v4/DHCP4: 198.51.100.6/24
                                 v6/DHCP6: 2001:db8::20c:29ff:fe78:6e4e/64
LAN (lan)       -> vmx1       -> v4: 10.6.0.1/24
                                 v6/t6: 2001:db8:1:eea0:20c:29ff:fe78:6e58/64

0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Disable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

Page Contents

- *1) Assign Interfaces*
- *2) Set interface(s) IP address*
- *3) Reset webConfigurator password*
- *4) Reset to factory defaults*
- *5) Reboot system*
- *6) Halt system*
- *7) Ping host*
- *8) Shell*
- *9) pfTop*
- *10) Filter Logs*

- *11) Restart webConfigurator*
- *12) PHP shell + pfSense tools*
- *13) Upgrade from console*
- *14) Enable/Disable Secure Shell (sshd)*
- *15) Restore recent configuration*
- *16) Restart PHP-FPM*

8.7.1 1) Assign Interfaces

This option restarts the **Interface Assignment** task, which is covered in detail in *Assign Interfaces* and *Manually Assigning Interfaces*. This menu option can create VLAN interfaces, reassign existing interfaces, or assign new ones.

8.7.2 2) Set interface(s) IP address

The script to set an interface IP address can set WAN, LAN, or OPT interface IP addresses, but there are also other useful features of this script:

- The firewall prompts to enable or disable DHCP service for an interface, and to set the DHCP IP address range if it is enabled.
- If the firewall GUI is configured for HTTPS, the menu prompts to switch to HTTP. This helps in cases when the SSL configuration is not functioning properly.
- If the anti-lockout rule on LAN has been disabled, the script enables the anti-lockout rule in case the user has been locked out of the GUI.

8.7.3 3) Reset webConfigurator password

This menu option invokes a script to reset the `admin` account password and status. The password is reset to the default value of `pfsense`.

The script also takes a few other actions to help regain entry to the firewall:

- If the GUI authentication source is set to a remote server such as RADIUS or LDAP, it prompts to return the authentication source to the Local Database.
- If the `admin` account has been removed, the script re-creates the account.
- If the `admin` account is disabled, the script re-enables the account.

8.7.4 4) Reset to factory defaults

This menu choice restores the system configuration to factory defaults. It will also attempt to remove any installed packages.

This action is also available in WebGUI at **Diagnostics > Factory Defaults**.

See *Resetting to Factory Defaults* for more details about how this process works.

8.7.5 5) Reboot system

This menu choice cleanly shuts down the firewall and restarts the operating system. There are several options which control what the firewall will do when rebooting. The choices offered by the reboot option are explained in *Reboot Methods*.

See also:

This action is also available in WebGUI at **Diagnostics > Reboot**, see *Rebooting the Firewall* for details.

8.7.6 6) Halt system

This menu choice cleanly shuts down the firewall and either halts or powers off, depending on hardware support.

Warning: The best practice is to **never** cut power from a running system. Halting before removing power is always the safest choice.

See also:

This action is also available in WebGUI at **Diagnostics > Halt System**. See *Halting and Powering Off the Firewall* for additional details.

8.7.7 7) Ping host

This menu option runs a script which attempts to contact a host to confirm if it is reachable by the firewall through a connected network. The script prompts the user for an IP address, and then the script sends that target host three ICMP echo requests.

The script displays output from the test, including the number of packets received, sequence numbers, response times, and packet loss percentage.

The script uses `ping` when given an IPv4 address or a hostname, and `ping6` when given an IPv6 address.

This is only a basic ping test. For more options, see *Ping Host* to run a similar test from the GUI.

8.7.8 8) Shell

This menu choice starts a command line shell.

Warning: A shell is very useful and very powerful, but also has the potential to be very dangerous.

Note: The majority of users do not need to touch the shell, or even know it exists.

Complex configuration tasks may require working in the shell, and some troubleshooting tasks are easier to accomplish from the shell, but there is always a chance of causing irreparable harm to the system.

Veteran FreeBSD users may feel slightly at home there, but there are many commands which are not present on pfSense software installations since unnecessary parts of the OS are removed for security and size constraints.

A shell started in this manner uses `tcsh`, and the only other shell available is `sh` . While it is possible to install other shells for the convenience of users, Netgate neither recommends nor supports using other shells.

8.7.9 9) pfTop

This menu option invokes `pftop` which displays a real-time view of the firewall states, and the amount of data they have sent and received. It can help pinpoint sessions currently using large amounts of bandwidth, and may also help diagnose other network connection issues.

See also:

See *Viewing States with pfTop* for more information on how to use pfTop.

8.7.10 10) Filter Logs

The **Filter Logs** menu option displays firewall log entries in real-time, in their raw form. The raw logs contain much more information per line than the log view in the WebGUI (**Status > System Logs**, **Firewall tab**), but not all of this information is easy to read.

Tip: For a simplified console view of the firewall logs in real time with low detail, use the following shell command:

```
clog -f /var/log/filter.log | filterparser.php
```

8.7.11 11) Restart webConfigurator

Restarting the webConfigurator will restart the system process that runs the GUI (`nginx`). In extremely rare cases the process may have stopped, and restarting it will restore access to the GUI.

If the GUI is not responding and this option does not restore access, invoke menu option 16 to **Restart PHP-FPM** after using this menu option.

8.7.12 12) PHP shell + pfSense tools

The PHP shell is a powerful utility that executes PHP code in the context of the running system. As with the normal shell, it is also potentially dangerous to use. This is primarily used by developers and experienced users who are intimately familiar with both PHP and the pfSense software code base.

Playback Scripts

There are several playback scripts for the PHP Shell that automate simple tasks or enable access to the GUI.

These scripts are run from within the PHP shell like so:

```
pfSense shell: playback scriptname
```

They may also be run from the command line:

```
# pfSsh.php playback scriptname
```

changepassword

This script changes the password for a user, and also prompts to reset the account properties if it is disabled or expired.

disablecarp / enablecarp

These scripts disable and enable CARP high availability functions, and will deactivate CARP type Virtual IP addresses. This action does not persist across reboots.

disablecarpmaint / enablecarpmaint

These scripts disable and enable CARP maintenance mode, which leaves CARP active but demotes this unit so the other node can assume control. This maintenance mode will persist across reboots.

disabledhcpd

This script removes all DHCP configuration from the firewall, effectively disabling the DHCP service and completely removing all of its settings.

disablerefercheck

This script disables the `HTTP_REFERER` check mentioned in *Browser HTTP_REFERER enforcement*. This can help gain access to the GUI if a browser session is triggering this protection.

enableallowallwan

This script adds an allow all rule for IPv4 and IPv6 to the WAN interface.

> **Warning:** Be extremely careful with this option, it is meant to be a **temporary** measure to gain access to services on the WAN interface of the firewall in situations where the LAN is not usable. Once proper access rules are put in place, remove the rules added by this script.

enablesshd

This script enables the SSH daemon, the same as the console menu option or GUI option.

externalconfiglocator

This script will look for a `config.xml` file on an external device, such as a USB thumb drive, and will move it in place for use by the firewall.

gatewaystatus

This script prints the current gateway status and statistics. It also accepts an optional parameter `brief` which prints only the gateway name and status, omitting the addresses and statistical data.

generateguicert

This script creates a new self-signed certificate for the firewall and activates it for use in the GUI. This is useful in cases where the previous certificate is invalid or otherwise not usable. It also fills in the certificate details using the firewall hostname and other custom information, to better identify the host.

gitsync

This complex script synchronizes the PHP and other script sources with files from the pfSense github repository. It is most useful on development snapshots to pick up changes from more recent commits.

> **Warning:** This script can be dangerous to use in other circumstances. Only use this under the direction of a knowledgeable developer or support representative.

If the script is run without any parameters it will print a help message outlining its use. More information can be found at *Using gitsync to Update pfSense Between Snapshots*.

installpkg / listpkg / uninstallpkg

These scripts interface with the package system in a similar way to the GUI. These are primarily used for debugging package issues, comparing information in `config.xml` compared to the package database.

pfanchordrill

This script recursively searches through `pf` anchors and prints any NAT or firewall rules it finds. This can help track down unexpected behavior in areas such as UPnP which rely on rules in anchors that are not otherwise visible in the GUI.

pftabledrill

This script prints the contents of all `pf` tables, which contain addresses used in firewall aliases as well as built-in system tables for features such as bogon network blocking, snort, and GUI/SSH lockout. This script is useful for checking if a specific IP address is found in any table, rather than searching individually.

removepkgconfig

This script removes all traces of package configuration data from the running `config.xml`. This can be useful if a package has corrupted settings or has otherwise left the packages in an inconsistent state.

removeshaper

This script removes ALTQ traffic shaper settings, which can be useful if the shaper configuration is preventing rules from loading or is otherwise incorrect and preventing proper operation of the firewall.

resetwebgui

This script resets the GUI settings for widgets, dashboard columns, the theme, and other GUI-related settings. It can return the GUI, particularly the dashboard, to a stable state if it is not functioning properly.

restartallwan

This script disables and re-enables each WAN-type interface, which reapplies the interface configuration.

restartdhcpd

This script stops and restarts the DHCP daemon.

restartipsec

This script rewrites and reloads the IPsec configuration for strongSwan.

svc

This script controls the services running on the firewall, similar to interacting with services at **Status > Services**.

The general form of the command is:

```
playback svc <action> <service name> [service-specific options]
```

The **action** can be `stop`, `start`, or `restart`.

The **service name** is the name of the services as found under **Status > Services**. If the name includes a space, enclose the name in quotes.

The **service-specific options** vary depending on the service, they are used to uniquely identify services with multiple instances, such as OpenVPN or Captive Portal entries.

Examples:

- Stop miniupnpd:

```
pfSsh.php playback svc stop miniupnpd
```

- Restart OpenVPN client with ID 2:

```
pfSsh.php playback svc restart openvpn client 2
```

- Start the Captive Portal process for zone "MyZone":

```
pfSsh.php playback svc start captiveportal MyZone
```

8.7.13 13) Upgrade from console

This menu option runs the `pfSense-upgrade` script to upgrade the firewall to the latest available version. This is operationally identical to running an upgrade from the GUI and requires a working network connection to reach the update server.

This method of upgrading is covered with more detail in *Upgrading using the Console*.

8.7.14 14) Enable/Disable Secure Shell (sshd)

This option toggles the status of the Secure Shell Daemon, `sshd`. This option works the same as the option in the WebGUI to enable or disable SSH.

8.7.15 15) Restore recent configuration

This menu option starts a script that lists and restores backups from the configuration history. This is similar to accessing the configuration history from the GUI at **Diagnostics > Backup/Restore** on the **Config History** tab (*Restoring from the Config History*).

This script can display the last few configuration files, along with a timestamp and description of the change made in the configuration, the user and IP address that made the change, and the config revision. This is especially useful if a recent configuration error accidentally prevented access to the GUI.

8.7.16 16) Restart PHP-FPM

This menu option stops and restarts the daemon which handles PHP processes for `nginx`. If the GUI web server process is running but unable to execute PHP scripts, invoke this option. Run this option in conjunction with **Restart webConfigurator** for the best result.

8.8 Resetting to Factory Defaults

The firewall configuration can be reset back to defaults, a process which also attempts to remove any installed packages. This reset can be performed in the GUI from **Diagnostics > Factory Defaults**, by using the console menu, or in some cases by using a hardware button.

In each case, the firewall will automatically reboot with a default configuration after the reset, which may require console access to resolve.

Note: This process does not remove any changes made to the file system, it only resets the configuration.

If system files have been corrupted or altered in an undesirable way, the best practice is to make a backup and reinstall from installation media.

8.8.1 Factory Default from the GUI

To reset the configuration to factory defaults using the GUI:

- Navigate to **Diagnostics > Factory Defaults**
- Review the items on the page which will be affected by the reset
- Click  **Factory Reset**
- Click **OK** to confirm the action and start the reset process

8.8.2 Factory Default from the Console

To reset the configuration to factory defaults using the console:

- Access the console menu locally or via SSH with an admin-level account (`admin`, `root`, or another privileged account using `sudo`).
- Enter the menu option which corresponds with **Reset to factory defaults** (e.g. `4`)
- Press `Enter`
- Enter the `y` to confirm the action
- Press `Enter` to start the reset process

8.8.3 Factory Default using a Hardware Button

On some appliances from Netgate, the reset button may be depressed with a paperclip or other similar object during the boot sequence.

> **Warning:** Reset button behavior varies by hardware. Check the appropriate product manual to confirm support and button behavior before attempting this procedure.

For most hardware which supports this feature, the procedure is similar:

- Apply power to the unit

- Depress the reset button after the initial POST sequence completes
- Hold the reset button in until the system LEDs turn off or the system reboots

The unit will reset the configuration to factory defaults and reboot again with that default configuration.

8.9 XML Configuration File

pfSense® software stores its settings in an XML format configuration file. All configuration settings including settings for packages are held in this one file. Run-time configuration files for services and firewall behavior are generated dynamically based on the settings held within this XML configuration file.

Those familiar with FreeBSD and related operating systems have found this out the hard way, when their changes to system configuration files were repeatedly overwritten by the firewall before they came to understand that pfSense software handles everything automatically.

The configuration file is stored at `/conf/config.xml` on the firewall.

8.9.1 Manually editing the configuration

A handful of configuration options are only available by manually editing the configuration file, though this isn't required in the vast majority of deployments. Some of these options are covered in other parts of this documentation where they are relevant. Additionally, for advanced administrators in rare cases large-scale or tricky changes may be easier to make by directly editing the configuration file.

Warning: Even for seasoned administrators it is easy to incorrectly edit the configuration file. Always keep backups and be aware that breaking the configuration will result in unintended consequences.

Edit a Backup

The safest and easiest method of editing the configuration file is to make a backup, edit the backup, and then restore:

- Navigate to **Diagnostics > Backup/Restore** in the GUI
- Download and save backup file
- Open the file in a text editor that properly understands UNIX line endings, and preferably an editor that has special handling for XML such as syntax highlighting. Do not use `notepad.exe` on Windows.
- Make changes to the configuration and save
- Navigate to **Diagnostics > Backup/Restore** in the GUI
- Restore the edited configuration

The firewall will automatically reboot as a part of the restoration process, and the new settings will be active afterward.

Edit In Place

Editing the configuration in-place is also possible in a variety of ways. The general procedure is:

- Edit `/conf/config.xml`
- Run `rm /tmp/config.cache` to clear the configuration cache
- Reboot, or use the GUI to save/reload whichever part of the firewall utilizes the edited settings

From the console or ssh, administrators familiar with the vi editor can use the `viconfig` command to edit the running configuration, and this command automatically clears the cache file after saving and exiting.

Other editors are available on the firewall, such as `ee` or in the GUI under **Diagnostics > Edit File** (*Editing Files on the Firewall*). Clear the cache file manually after using one of these other methods, either using the shell or **Diagnostics > Command Prompt** (*Command Prompt*).

8.10 pfSense® Plus Software Registration

The pfSense Plus Software Registration page is located at **System > Register**. This page activates features in pfSense Plus software installations on hardware and virtual machines not purchased from Netgate. The page also activates older Netgate® hardware purchased with Factory Edition (FE) pfSense software before the Netgate Device ID (NDI) was introduced.

The registration process requires an activation token supplied by Netgate. This token is generated when purchasing pfSense Plus software or via Netgate TAC for older Netgate hardware.

For more information about pfSense Plus software, or to purchase pfSense Plus software, visit Netgate Store.

Note: Registration is free for hardware purchased from Netgate with pfSense Plus software or the older Factory Edition of pfSense software. Most hardware is pre-registered and does not require activation. To activate hardware which is not automatically recognized, submit a request to Netgate TAC along with the serial number and NDI for the device at https://go.netgate.com. The serial number and NDI are displayed on the dashboard in the **System Information** widget, and may also be on a sticker located on the bottom of the device.

The current registration status is shown on the dashboard in the **Netgate Services and Support** widget, and is also indicated on **System > Register**.

The text on the registration page varies depending on the current registration status and availability. The page also displays errors encountered during the activation process, such as not being able to contact the registration server.

8.10.1 Registration Process

To register an installation of pfSense Plus software with Netgate:

- Obtain a pfSense Plus software activation token from Netgate
- Navigate to **System > Register** on the firewall
- Enter the **Activation Token**
- Click **Register**

See also:

- *Basic Firewall Configuration Example*
- *Troubleshooting Clock Issues*

- *Troubleshooting*
- *Troubleshooting Access when Locked Out of the Firewall*
- *Troubleshooting Time Zone Configuration*

Most pfSense® software configuration is performed using the web-based GUI. There are a few tasks that may also be performed from the console, whether it be a monitor and keyboard, over a serial port, or via SSH.

8.11 Connecting to the GUI

To reach the GUI, follow this basic procedure:

- Connect a client computer to the same network as the LAN interface of the firewall. This computer may be directly connected with a network cable or connected to the same switch as the LAN interface of the firewall.

  By default, the LAN IP address of a new installation of pfSense software is `192.168.1.1` with a `/24` mask (`255.255.255.0`), and there is also a DHCP server running. If a client computer is set to use DHCP, it should obtain an address in the LAN subnet automatically.

- On the client computer, open a web browser such as Firefox, Safari, or Chrome and navigate to https://192.168.1.1.

  The GUI listens on HTTPS by default, but if the browser attempts to connect using HTTP, it will be redirect by the firewall to the HTTPS port instead.

- Enter the default credentials in the login page:

  username `admin`

  password `pfsense`

In some cases additional steps may be necessary before the client computer can reach the GUI.

Warning: If the default LAN subnet conflicts with the WAN subnet, the LAN subnet must be changed before connecting it to the rest of the network. Attempting to access the GUI in this situation is unpredictable and unlikely to work until the conflict is resolved.

The LAN IP address may be changed and DHCP may be disabled using the console:

- Open the console (VGA, serial, or using SSH from another interface)
- Choose option `2` from the console menu
- Enter the new LAN IP address, subnet mask, and specify whether or not to enable DHCP.
- Enter the starting and ending address of the DHCP pool if DHCP is enabled. This can be any range inside the given subnet.

Note: When assigning a new LAN IP address, it cannot be in the same subnet as the WAN or any other active interface. If there are other devices already present on the LAN subnet, it also cannot be set to the same IP address as an existing host.

If the DHCP server on the firewall is disabled, client computers on LAN must have a statically configured IP address in the LAN subnet, such as `192.168.1.5`, with a subnet mask that matches the one given to the firewall, such as `255.255.255.0`.

CHAPTER

NINE

BACKUP AND RECOVERY

9.1 Making Backups in the GUI

Making a backup in the GUI is simple:

- Navigate to **Diagnostics > Backup & Restore**
- Set any desired options, or leave the options at their default values.
- Click **Download Configuration as XML** (Figure *GUI Backup*).

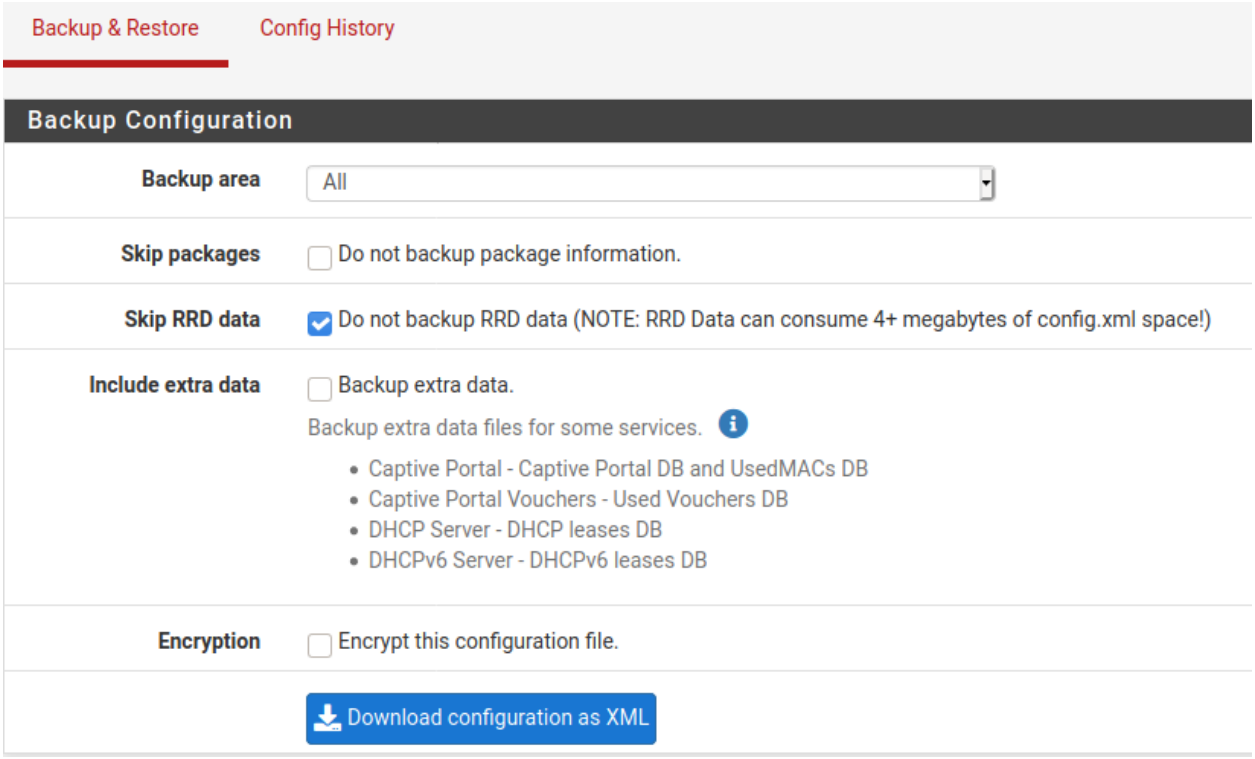


Fig. 1: GUI Backup

The web browser will then prompt to save the file somewhere on the PC being used to view the GUI. It will be named `config-<hostname>-<timestamp>.xml`, but that may be changed before saving the file.

9.1.1 Backup Options

When performing a backup, GUI options are available to control what is contained within the backup file.

Backup Area Limits the backup contents to a single configuration area, rather than a complete configuration backup.

The default behavior is to inclue all areas in the backup.

Note: When restoring a configuration containing only a single area, the **Restore area** value must be set to match.

Skip Packages Controls whether or not the backup will contain installation data and settings for packages. Omitting this data from a backup can be a useful way to quickly remove all traces of packages from a configuration when troubleshooting.

> **Warning:** After restoring a configuration without package data all packages must be reinstalled and reconfigured.

The default is *unchecked* so that all package data is included in the backup.

Skip RRD Data Controls whether or not the backup will contain an exported copy of data used to generate *monitoring graphs*. When restoring a backup containing RRD data, the graph data is also restored.

The default is *checked* which omits the RRD data from the backup as it significantly increases the size of backup files.

Include Extra Data Controls whether or not the backup file will include additional optional data. This includes Captive Portal databases and DHCP lease databases. These databases are volatile. While the data can be useful for transferring to new hosts or for frequent backups, it is not as useful for long-term backups.

The default is *unchecked* which omits this extra data from the backup as it can significantly increase the size of backup files.

Encryption Controls whether or not the backup file is encrypted before download.

When set, the GUI presents **Password** and confirmation fields, the contents of which are used by pfSense® software to encrypt the backup file with AES- 256.

The default behavior is *unchecked* which creates clear text XML backup files.

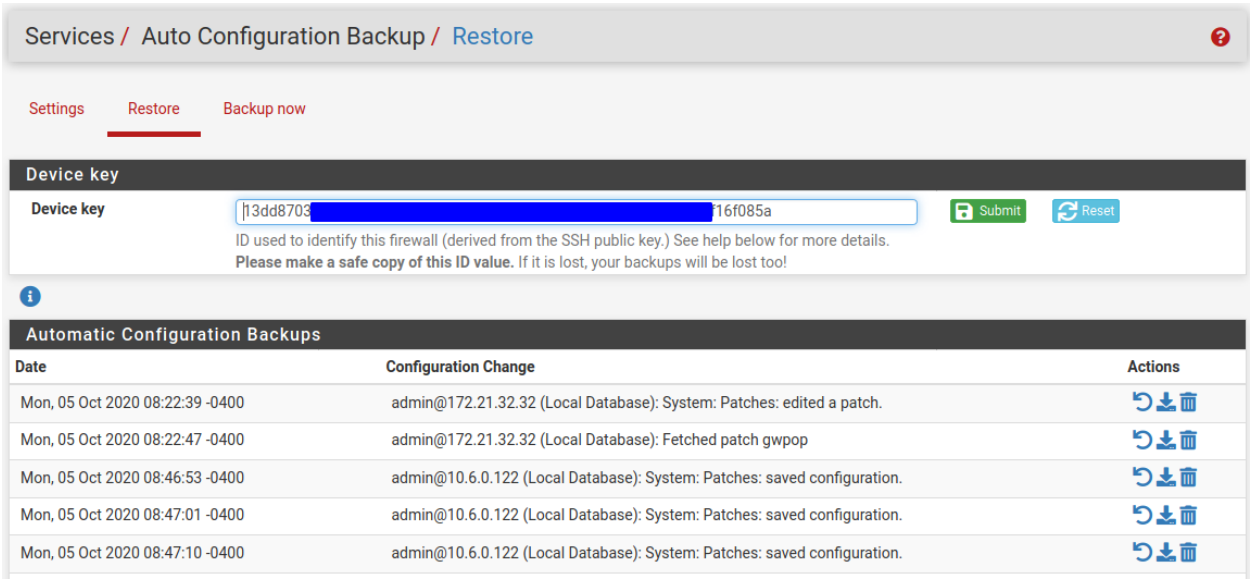
9.2 Using the AutoConfigBackup Service

Automatic Configuration Backup (AutoConfigBackup, ACB) is available as a core component of pfSense® software, no package required.

9.2.1 Functionality and Benefits

When a change is made to the configuration on a firewall, AutoConfigBackup automatically encrypts the contents with the passphrase entered in the AutoConfigBackup settings and then uploads the backup over HTTPS to Netgate servers. This gives instant, secure offsite backups of a firewall with no user intervention.

Only the most recent 100 encrypted configurations for each device are retained on Netgate servers.



9.2.2 Encryption Password

Before the configuration is transmitted to Netgate servers, the firewall encrypts the backup using the AES-256-CBC algorithm and a password created by the firewall administrator on the **Settings** tab (*Configuration*). This password is only used locally by AutoConfigBackup and is not transmitted to remote servers.

When restoring a backup from the list of available remote backups, the contents are downloaded and then decrypted with the configured encryption password.

> **Warning: Keep a careful record of the encryption password!**
>
> If the password is lost, the backup contents **cannot be recovered**. The password is private and only known to the local firewall. Neither Netgate nor anyone else will be able to assist in reading the encrypted backups without the password.

9.2.3 Device Key

To identify a specific firewall, an unique identifier is required to save or restore a backup configuration. ACB uses the SHA256 hash of the SSH public key on the firewall for this purpose.

The device key is located on the **Services > Auto Configuration Backup** menu item, under the **Restore** and **Backup now** tabs.

Warning: Keep a careful record of this Device Key!

If the **Device Key** of a firewall is lost, there is a chance it can be recovered. The **Settings** page allows the entry of a **Hint** which is stored in the data store alongside the encrypted backup entries. If the hint is distinct, the Netgate support team *may* be able to use it to recover the device key. Do not count on this though!

9.2.4 Configuration

To adjust the settings navigate to **Services > Auto Config Backup**, **Settings** tab.

Configuring AutoConfigBackup

Enable ACB When checked, ACB is active and will make automatic configuration backups.

Backup Frequency Select when ACB will create backups

On Every Configuration Change When selected, ACB will perform a backup on every significant configuration change.

Note: Some minor configuration changes are safely ignored if they do not impact functionality.

On a Regular Schedule Enables **Schedule** controls to perform timed backups instead of performing a backup on every change. This can be more efficient on systems with many frequent changes.

Schedule Controls the **Hours** of the day, **Day** of the month, **Month** of the year, and **Day of the week** on which backups are performed using the standard cron format.

Note: This control is only visible when **Backup Frequency** is set to *On a Regular Schedule*.

Encryption Password/Confirm The password used by ACB to encrypt the backup, as described in *Encryption Password*.

Hint/Identifier An optional hint which will be stored as plain text metadata along with the encrypted configuration. This hint may allow Netgate TAC to locate the device key if it is lost.

Manual Backups to Keep Up to `50` manual backups may be retained, which are not automatically overwritten by automatic backups. These manual backups still count against the `100` backup limit.

Testing Backup Functionality

- Make a change to force a configuration backup, such as editing and saving a firewall or NAT rule.
- Click **Apply Changes**
- Navigate to **Services > Auto Config Backup**, **Restore** tab
- Look for the new backup in the list

Manually Backing Up

Manual backups should be made before an upgrade or a series of significant changes, as it will store a backup specifically showing the reason, which then makes it easy to restore if necessary. Since each configuration change triggers a new backup, when a series of changes is made it can be difficult to know where the process started.

To force a manual backup of the configuration:

- Navigate to **Diagnostics > AutoConfigBackup**
- Click the **Backup Now** tab at the top
- Enter a **Revision Reason**
- Click **Backup**

Tip: Take a manual backup prior to upgrading to a new pfSense software release, and name the backup so the reason the backup was made is clear.

Restoring a Configuration

To restore a configuration:

- Navigate to **Diagnostics > AutoConfigBackup**
- Click the **Restore** tab at the top
- Locate the desired backup in the list
- Click  to the right of the configuration row

The firewall will download the configuration specified from the AutoConfigBackup server, decrypt it with the **Encryption Password**, and restore it.

Warning: By default, the firewall **will not** initiate a reboot. Depending on the configuration items restored, a reboot may not be necessary. For example, firewall and NAT rules are automatically reloaded after restoring a configuration.

After restoring, a the GUI presents a prompt offering to reboot. If the restored configuration changes anything other than the NAT and firewall rules, choose **Yes**.

9.2.5 Bare Metal Restoration

If the disk in the firewall fails or if the SSH key changes due to a re-installation of pfSense software, the ACB service can restore a backup from the previous installation as long as the *Device Key* and the *Encryption Password* of the previous installation are both known.

- Replace the failed disk
- Install pfSense software on the new disk
- Configure LAN and WAN
- Navigate to **Diagnostics > AutoConfigBackup**, **Settings** tab
- Set the **Encryption Password** to match the previous installation

- Navigate to the **Restore** tab
- Paste the old device key into the **Device Key** field
- Click the **Submit** button

This temporarily allows ACB to display a list of backups for an alternate **Device Key**.

Click  **Reset** to restore the native ID for this firewall.

Once the firewall has been rebooted, it will be running with the configuration backed up before the failure.

9.2.6 Checking the AutoConfigBackup Status

The status of an AutoConfigBackup run cay be checked by reviewing the list of backups shown on the **Restore** tab. This list is pulled from the AutoConfigBackup servers. If the backup is listed there, it was successfully created.

If a backup fails, an alert is logged, and it will be visible as a notice in the WebGUI.

9.3 Alternate Remote Backup Techniques

The easiest choice to backup pfSense® software is the free *Using the AutoConfigBackup Service* service. Rest easy knowing it is taking care of handling remote backups automatically without needing to worry. Sit back, have a cup of coffee, and read on for alternate techniques.

The other techniques in this document may also be used to perform backups remotely, but each method has its own security issues which may rule out their use. For starters, several of these techniques do not encrypt the configuration, which may contain sensitive information. This can result in the raw configuration being transmitted over an unencrypted, untrusted link. If one of these techniques must be used, it is best to do so from a non-WAN link (LAN, DMZ, etc.) or across a VPN. Access to the storage media holding the backup must also be controlled, if not encrypted.

9.3.1 Pull

Pulling the configuration means to use a remote client to "pull" the configuration off of the firewall. The methods described below both accomplish the same goal using different utilities.

Pull with wget

The configuration may be retrieved from a remote system by using `wget` , and this process can be scripted with cron or by other means. Even when using HTTPS, this is not a truly secure transport mode since certificate checking is disabled to accommodate self-signed certificates, enabling man-in-the-middle attacks. When running backups with `wget` across untrusted networks, use HTTPS with a certificate that can be verified by `wget`.

The wget command must be split into multiple steps to handle the login procedure and backup download while also accounting for CSRF verification.

For a firewall running HTTPS with a self-signed certificate, the commands are as follows:

- Fetch the login form and save the cookies and CSRF token:

```
$ wget -qO- --keep-session-cookies \
  --save-cookies cookies.txt \
  --no-check-certificate \
  https://192.168.1.1/diag_backup.php \
  | grep "name='__csrf_magic'" \
  | sed 's/.*value="\(.*\)".*/\1/' > csrf.txt
```

- Submit the login form along with the first CSRF token and save the second CSRF token (can't reuse the same file) – now the script is logged in and can take action:

```
$ wget -qO- --keep-session-cookies --load-cookies cookies.txt \
  --save-cookies cookies.txt --no-check-certificate \
  --post-data "login=Login&usernamefld=admin&passwordfld=pfsense&__csrf_magic=
↪$(cat csrf.txt)" \
  https://192.168.1.1/diag_backup.php \
  | grep "name='__csrf_magic'" \
  | sed 's/.*value="\(.*\)".*/\1/' > csrf2.txt
```

- Submit the download form along with the second CSRF token to save a copy of `config.xml`:

```
$ wget --keep-session-cookies --load-cookies cookies.txt --no-check-certificate \
  --post-data "download=download&donotbackuprrd=yes&__csrf_magic=$(head -n 1␣
↪csrf2.txt)" \
  https://192.168.1.1/diag_backup.php -O config-router-`date +%Y%m%d%H%M%S`.xml
```

Note: The behavior of variable expansion and other aspects of the commands may vary by shell. This example uses `bash` for the client shell.

Replace the username and password with the credentials for the firewall, and the IP address is whichever IP address is reachable from the client performing the backup, and using HTTP or HTTPS to match the firewall GUI. To backup the RRD files, omit the `&donotbackuprrd=yes` parameter from the last command.

The system performing the backup will also need access to the GUI, so adjust the firewall rules accordingly. Performing this over the WAN is not recommended. At a minimum, use HTTPS and restrict access to the GUI to a trusted set of public IP addresses. It is preferable to do this locally or over a VPN.

Using cURL

For those who prefer to use cURL, the following example accomplishes the same goal:

- Fetch the login form and save the cookies and CSRF token:

```
$ curl -L -k --cookie-jar cookies.txt \
    https://192.168.1.1/ \
    | grep "name='__csrf_magic'" \
    | sed 's/.*value="\(.*\)".*/\1/' > csrf.txt
```

- Submit the login form to complete the login procedure:

```
$ curl -L -k --cookie cookies.txt --cookie-jar cookies.txt \
    --data-urlencode "login=Login" \
    --data-urlencode "usernamefld=admin" \
    --data-urlencode "passwordfld=pfsense" \
    --data-urlencode "__csrf_magic=$(cat csrf.txt)" \
    https://192.168.1.1/ > /dev/null
```

Now the script is logged in and can perform actions!

- Fetch the target page to obtain a new CSRF token:

```
$ curl -L -k --cookie cookies.txt --cookie-jar cookies.txt \
     https://192.168.1.1/diag_backup.php  \
     | grep "name='__csrf_magic'"   \
     | sed 's/.*value="\(.*\)".*/\1/' > csrf.txt
```

- Download the backup:

```
$ curl -L -k --cookie cookies.txt --cookie-jar cookies.txt \
     --data-urlencode "download=download" \
     --data-urlencode "donotbackuprrd=yes" \
     --data-urlencode "__csrf_magic=$(head -n 1 csrf.txt)" \
     https://192.168.1.1/diag_backup.php > config-router-`date +%Y%m%d%H%M%S`.xml
```

Note: The behavior of variable expansion and other aspects of the commands may vary by shell. This example uses `bash` for the client shell.

9.3.2 Push with SCP

The configuration file can also be pushed from the firewall to another host with `scp`. Using `scp` to push a one-time backup by hand can be useful, but using it in an automated fashion carries risks. The command line for `scp` will vary depending on the system configuration, but will be close to the following:

```
$ scp /cf/conf/config.xml \
    user@backuphost:backups/config-`hostname`-`date +%Y%m%d%H%M%S`.xml
```

Pushing the configuration in an automated manner requires the firewall administrator to generate an SSH key without a passphrase. Due to the insecure nature of a key without a passphrase, generating such a key is left as an exercise for the reader. This adds risk due to the fact that anyone with access to that file has access to the designated account, though because the key is kept on the firewall where access is restricted, it isn't a considerable risk in most scenarios. Ensure the remote user is isolated and has little to no privileges on the destination system.

A chrooted `scp` environment may be desirable in this case. The `scponly` shell is available for most UNIX platforms which allows SCP file copies but denies interactive login capabilities. Some versions of OpenSSH have chroot support built in for sftp (Secure FTP). These steps greatly limit the risk of compromise with respect to the remote server, but still leave the backed up data at risk. Once access is configured, a cron entry could be added to the firewall to invoke `scp`.

A summary of the setup is as follows:

- Generate an ssh key for the root user on the firewall *without a passphrase*. (Warning: dangerous!)
- Add a user to a remote system, and add the new public key to its `~/.ssh/authorized_keys` file
- Create a cron job on the firewall that would copy `/cf/conf/config.xml` to the remote system with `scp`

9.3.3 Basic SSH backup

Similar to the `scp` backup, there is another method that will work from one UNIX system to another. This method does not invoke the SCP/SFTP layer, which in some cases may not function properly if a system is already in a failing state:

```
$ ssh root@192.168.1.1 cat /cf/conf/config.xml > backup.xml
```

When executed, that command will yield a file called `backup.xml` in the current working directory that contains the remote firewall configuration. Automating this method using cron is also possible, but this method requires an SSH key without as passphrase on the host performing the backup. This key will enable administrative access to the firewall, so it must be tightly controlled. (See *Secure Shell (SSH)* for details.)

9.4 Restoring from Backups

Backups are not useful without a means to restore them, and by extension, test them. Several means for restoring configurations are available in pfSense® software. Each method has the same end result: a running firewall identical to when the backup was made.

9.4.1 Backup Compatibility

The version of pfSense Plus or pfSense CE software is not as important as the **Configuration Revision** number when determining backup compatibility. Differences in the configuration revision number indicate changes in the format of the configuration data which makes them not directly compatible.

See also:

There is a list of software versions and their corresponding configuration revision numbers at *Versions of pfSense software and FreeBSD*.

Backups using the same configuration revision can be restored as-is, both for complete configuration backups and partial (section-based) backups.

Complete backups with a lower configuration revision **can** be restored to a current version. The upgrade code will adjust the values in the configuration to convert it into a current format.

Partial (section-based) backups **cannot** be restored if they were taken on a version with a different configuration revision, as there is no mechanism for the upgrade code to handle partial backups.

Backups with a higher configuration revision **cannot** be restored to an older version. There is no mechanism to downgrade a configuration as the older version will have no knowledge of changes which happened in future versions of the software.

Restoring between pfSense CE and pfSense Plus or vice versa may work in many cases, but results depend upon the target hardware and version. For example, restoring to pfSense Plus on hardware with an integrated Ethernet switch may require manual adjustments. Contact Netgate TAC for specific guidance.

9.4.2 Restoring with the GUI

The easiest way for most users to restore a configuration is by using the GUI:

- Navigate to **Diagnostics > Backup & Restore**
- Locate the **Restore Backup** section (Figure *GUI Restore*).
- Select the area to restore, or leave at the default selection for a complete backup.

Note: This value must match the **Backup area** chosen when creating the backup.

- Click **Browse**
- Locate the backup file on the local PC
- Click **Restore Configuration**

The firewall will then apply the configuration and reboot with the settings obtained from the backup file.

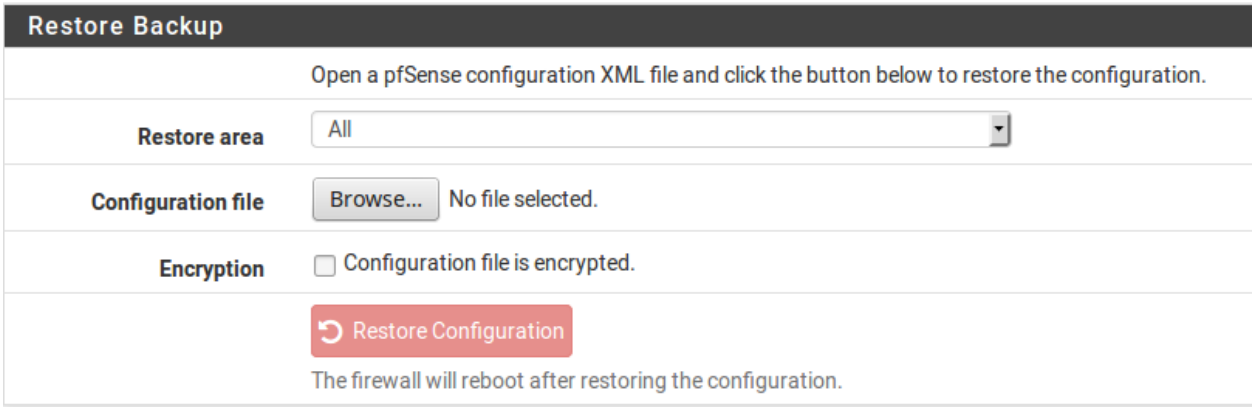


Fig. 2: GUI Restore

While easy to work with, this method has prerequisites when dealing with a full restore to a new installation. First, it would need to be done after the new target system is fully installed and running. Second, it requires an additional PC connected to a working network or crossover cable behind the firewall being restored.

Restore Options

Restore Area Restores a backup containing only a single configuration area, rather than a complete configuration backup.

> **Warning:** Restoring a single area does not trigger a reboot nor does it cause any part of the configuration to be reapplied. To ensure the restored configuration area is active, issue a reboot or manually refresh the configuration for the relevant area after restore (e.g. edit/save/apply on a page, issue a filter reload, etc).

> **Warning:** When restoring a single area, the area being restored must be from the **same** version. Single areas do not support running upgrade code on the configuration, and thus cannot be adjusted if the format of the area changed from a previous version.

> **Warning:** This does not restore one area from a full backup, the backup file must only contain the area to restore.

Note: This value must match the **Backup area** chosen when creating the backup.

Configuration File A **Browse** button to select a backup file to upload and restore.

Preserve Switch Configuration This option is available on Netgate hardware with integrated switches. When set, the current active switch configuration will be copied into the restored configuration, preserving it for later use. This makes it easier to restore a configuration from hardware without an integrated switch.

Note: This only copies the integrated switch configuration, and does not copy VLAN or LAGG interface entries which may be relevant to using the switch. This behavior is safer, as the configuration being restored may also contain important configuration data in those areas.

Encryption When set, a **Password** field is presented, the contents of which is used by the firewall to decrypt the contents of the backup file before restoring the configuration.

9.4.3 Restoring from the Config History

For minor problems, using one of the internal backups on the firewall is the easiest way to back out a change. The previous `30` configurations are stored in the **Configuration History**, along with the current running configuration.

Each row in the configuration history list shows the date the configuration file was made, the configuration version, the user and IP address of a person making a change in the GUI, the page that made the change, and in some cases, a brief description of the change that was made. The action buttons to the right of each row show a description of what they do when the mouse pointer is hovered over the button.

To restore a configuration from the history:

- Navigate to **Diagnostics > Backup & Restore**
- Click the **Config History** tab (Figure *Configuration History*)
- Locate the desired backup in the list
- Click  to restore that configuration file

Restoring a configuration with this method does not initiate an automatic reboot. Minor changes do not require a reboot, though reverting some major changes will.

If a change was only made in one specific section, such as firewall rules, trigger a refresh in that area of the GUI to enable the changes. For firewall rules, a filter reload would be sufficient. For OpenVPN, edit and save the VPN instance. The necessary actions to take depend on the changes in the restored configuration, but the best way ensure that the full configuration is active is to reboot.

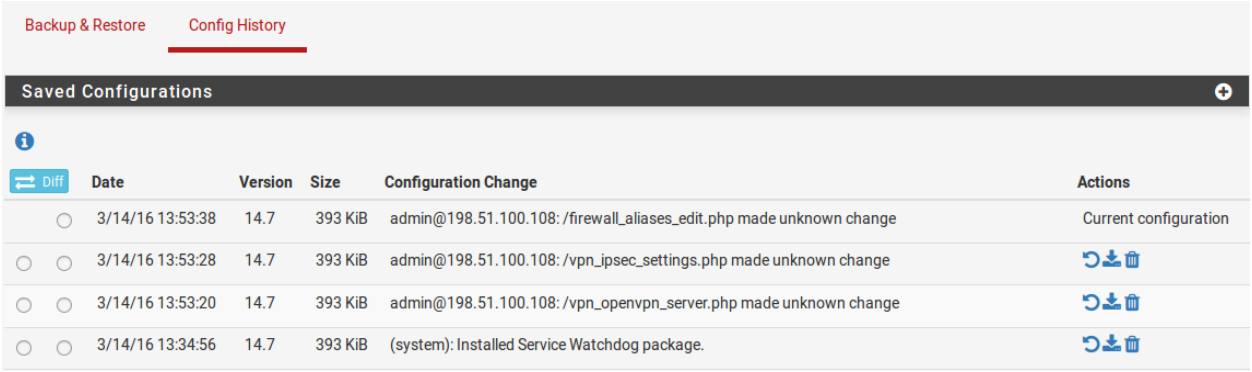


Fig. 3: Configuration History

If necessary, reboot the firewall with the new configuration by going to **Diagnostics > Reboot System** and click **Yes**.

Previously saved configurations may be deleted by clicking  , but do not delete them by hand to save space; the old configuration backups are automatically deleted when new ones are created. It is desirable to remove a backup from a known-bad configuration change to ensure that it is not accidentally restored.

A copy of the previous configuration may be downloaded by clicking  .

Configuration Backup Cache Settings

The amount of backups stored in the configuration history may be changed if needed.

- Navigate to **Diagnostics > Backup & Restore**
- Click the **Config History** tab
- Click  at the right end of the **Configuration Backup Cache Settings** bar to expand the settings
- Enter the new number of configurations to retain in the **Backup Count** field
- Click **Save**

Along with the configuration count, the page also displays the amount of space consumed by the backup cache.

Config History Diff

The differences between any two configuration files may be viewed in the **Config History** tab. To the left of the configuration file list there are two columns of radio buttons. Use the leftmost column to select the *older* of the two configuration files, and then use the right column to select the *newer* of the two files. Once both files have been selected, click **Diff** at either the top or bottom of the column.

Console Configuration History

The configuration history is also available from the console menu as option `15`, **Restore Recent Configuration**. The menu selection will list recent configuration files and offer to restore one. This is useful if a recent change has locked administrators out of the GUI or taken the firewall off the network.

9.4.4 Restoring by Mounting the Disk

Attaching the disk from an installation of pfSense software to a computer running FreeBSD enables the drive to be mounted by the FreeBSD host and a new configuration may be copied directly onto the installed system, or a configuration file from a failed system may be copied off.

Note: This can also be performed on a separate installation of pfSense in place of a computer running FreeBSD, but do not use an active production firewall for this purpose. Instead, use a spare or test firewall.

The `config.xml` file is kept in `/cf/conf/`, but the difference is in the location where this directory resides. This is part of the root slice (typically `da0p2`). The drive and partition name will vary depending on disk type and position in the host.

9.4.5 Encrypted Configuration files

The GUI can automatically determine the correct decryption method when restoring an encrypted configuration backup file, whether it's from a current version or an older version. When restoring an encrypted configuration file, check **Configuration file is encrypted** then enter the password in the **Password** field, and restore as usual from there.

Encrypted configuration files can be manually decrypted using the correct password for offline inspection.

The method used to encrypt configuration files changed in version 2.5.0, so use the method appropriate for the version which generated the encrypted configuration file. In either case, replace `<PASSWORD>` with the appropriate password string, and change the filenames as needed.

2.5.0 and later:

```
grep -v "config.xml" config-encrypted.xml | base64 -d | \
  openssl enc -d -aes-256-cbc -out dencryptedfile.xml \
  -pass pass:<PASSWORD> -salt -md sha256 -pbkdf2
```

Older versions:

```
grep -v "config.xml" config-encrypted.xml | base64 -d | \
  openssl enc -d -aes-256-cbc -out dencryptedfile.xml \
  -pass pass:<PASSWORD> -salt -md md5
```

9.5 Automatically Restore Configuration During Installation

In addition to restoring through the GUI, pfSense® software supports methods which restore a configuration to a new setup without going through all the trouble of setting up a client and restoring using a web browser.

These methods are significantly easier than reconfiguring the LAN and restoring via the network, especially in complex environments. The firewall will start up using the restored configuration immediately without needing intermediate steps.

- *Recover config.xml From Existing Installation*
- *Restore Configuration from USB During Install*
- *Restore using the External Configuration Locator (ECL)*

9.5.1 Recover config.xml From Existing Installation

The installer has a **Recover config.xml** option which reads the configuration file from an existing installation before starting the install process and puts it back in the exact same location when it finishes. This makes the feature useful for upgrades, filesystem changes, or any other situation requiring a reinstallation on the same disk.

Note: The **Recover config.xml** option works on installations using either UFS or ZFS.

- Take a backup of the configuration before starting, if possible, in case this procedure does not work as expected
- Boot a pfSense software installation image
- Choose **Recover config.xml** when the option appears
- Select the existing installation drive (e.g. `ada0`)

  The selection list shows the disk name, size, and filesystem type which is typically enough to identify the disk
- Wait a moment while the recovery process happens

  The recovery process attempts to repair the filesystem on the disk up to 10 times, then mounts the disk and looks for the existing configuration file. If it is able to find and read the configuration file, the recovery process copies it to a temporary RAM disk during the installation process.

  Note: The recovery process only briefly displays its output, so it can be difficult to spot whether it succeeded or failed. If the process fails, the configuration either is not there or it was not recoverable. Either way, proceeding is safe as it is unlikely the `config.xml` would be recovered from the drive by other means.
- Proceed through the installation as usual

At the end of the installation, the installer automatically copies the configuration from the temporary RAM disk back to the target disk before rebooting.

The firewall will boot off the target disk with the configuration restored by the installer already in place. The firewall will reinstall packages automatically in the background.

9.5.2 Restore Configuration from USB During Install

As part of the installation routine, the installer checks for an existing configuration on a USB drive formatted as FAT or FAT32. If the installer can locate and read a configuration file, it copies the file to the target disk.

> **Warning:** This feature does not support drives formatted with exFAT.
>
> For this feature to work correctly, the USB drive must contain a partition table and it must not be formatted as a raw device.

Tip: The pfSense software memstick installation image contains a FAT partition which the installer can use for this purpose. If the partition is not visible on the workstation which wrote the memstick image, remove and reinsert the USB drive.

- On a FAT/FAT32 formatted USB drive, make a directory called `conf`
- Copy a backup configuration file to the `conf` directory
- Rename the backup to `config.xml`

  Example: If the USB drive is `E:`, the full path would be `E:\conf\config.xml`

  Note: The installer also looks for `config.xml` in the root directory of the drive, but the best practice is to place the file in the `conf` directory.

- Unmount/eject the USB drive, remove it, then plug it into the firewall
- Boot the install media (Memstick, disc, etc)
- Install to the target disk
- Reboot the firewall
- Remove the USB drive only **AFTER** the firewall has begun to reboot

  > **Warning:** If the USB drive is removed too early, it may still be mounted and the system **will panic**!

- Remove the install media as well at this point

The firewall will boot off the target disk with the restored configuration.

9.5.3 Restore using the External Configuration Locator (ECL)

pfSense software also includes a feature called the External Configuration Locator, or ECL for short. The ECL process runs at boot time to, as the name implies, locate configuration files on external storage. If the ECL finds a configuration file, it copies that file to the firewall disk, replacing any existing configuration.

Note: The ECL runs on every boot, so its use is not limited to fresh installations.

This procedure is nearly identical to the method in *Restore Configuration from USB During Install*, but the USB disk containing the configuration does not need to be present during the installation. The same warnings from that procedure also apply here.

- On a FAT, FAT32, or UFS formatted USB drive, make a directory called `config`
- Copy a backup configuration file to the `config` directory
- Rename the backup to `config.xml`

  Example: If the USB drive is `E:`, the full path would be `E:\config\config.xml`.

Note: The ECL also looks for `config.xml` in the root directory of the drive, but the best practice is to place the file in the `config` directory.

- Unmount/eject and remove the USB drive
- Install pfSense software as usual

  This is optional, since the ECL runs on existing installations.
- Reboot the firewall
- Insert the USB drive containing the configuration while the firewall boots and the ECL will read in the configuration file from there

---

Note: USB drives which only contain files can be inserted before the firewall boots. Bootable USB drives, such as the installation memstick, should not be inserted until after the firewall has started to boot from its own disk. This behavior will vary by target device and its boot preferences. Monitor the console to find the appropriate timing.

Timing is also affected by the speed of the device. Slower systems may not mount the USB drive before the ECL runs.

- Wait for the firewall to complete the boot process
- Check that the configuration was loaded properly

  If the configuration did not load as expected, check the file location and name on the USB drive, and check the timing of when the USB drive was present during the boot process, then start over. Monitor the console for details.
- Remove the USB drive once the correct configuration file is in place

If this is the first boot post-installation, then this process also triggers reinstallation of packages listed in the restored configuration.

Warning: This procedure will copy the `config.xml` file from the USB drive to the target drive at **every** boot. However, the running firewall **will not** copy its own configuration back to the USB drive. Thus, leaving the drive inserted in the firewall will result in losing **all** configuration changes not present in the configuration file on the USB drive.

9.6 Restoring a Configuration File to a Different Version

Configurations are specific to a given version of pfSense® software. The configuration is the same on all platforms and architectures using the same version of pfSense software. The version of FreeBSD used is not relevant.

Generally speaking, a complete older configuration version can always be restored to a newer release of pfSense software. The firewall will upgrade the configuration as needed provided that has the **entire** configuration and not a partial copy.

A newer configuration **cannot** be restored to an older release that had a different configuration version. Certain releases of pfSense software had the same configuration version, and restoring between those is possible, but still not recommended. See *Versions of pfSense software and FreeBSD* to see which configuration versions were used on specific releases.

A configuration **section** or partial configuration cannot be restored between different configuration versions. It may work by pure luck, but often there are configuration format differences that require changes to be made to the older configuration. These changes are automatic if a complete configuration is restored. If a partial restore is required, perform a full upgrade in a test VM or lab and then copy the needed section out of the resulting `config.xml` post-upgrade.

9.7 Caveats and Gotchas

While the configuration XML file kept by pfSense® software includes all of the settings, it does not include any changes that may have been made to the system by hand, such as manual modifications of source code. Additionally some packages require extra backup methods for their data.

The configuration file may contain sensitive information such as VPN keys or certificates, and passwords (other than the admin password) in plain text. Some passwords must be available in plain text during run time, making secure hashing of those passwords impossible (*Password Storage Security Policies*). Hence backup copies of these files must also be protected in some way. If they are stored on removable media, take care with physical security of that media and/or encrypt the drive.

If the GUI must be used over the WAN without a VPN connection, at least use HTTPS. Otherwise, a backup is transmitted in the clear, including any sensitive information inside that backup file. We strongly recommend using a trusted network or encrypted connection.

9.8 Password Storage Security Policies

Sensitive data such as PPPoE/PPTP client, PPTP VPN, DynDNS passwords as well as remote authentication servers RADIUS (shared secret), LDAP (bind user password), and IPsec shared secrets, among others, appear in plain text or with reversible Base64 encoding in the pfSense® software configuration file, **config.xml**. This is a deliberate design decision in m0n0wall that has been carried over here.

Since the firewall cannot prompt the user for a password each time it is required, the implementations of affected areas require plain text passwords to operate. pfSense software could, of course, use some snake oil encryption on those passwords, but that would only create a false sense of security. Any encryption applied to the passwords could be reversed by anyone with access to the source code (i.e. everybody). Hashes like SHA256 cannot be used where the plain text password is needed at a later stage, unlike for the system password, which is only stored as a hash.

By leaving the passwords in plain text, it is very clear that **config.xml** deserves to be stored in a secure location (and/or encrypted with one of the countless programs out there). Any sort of hashing used would not be secure, and would be dangerous because it would give the impression of security where none exists.

See also:

- *Backup Files and Directories with the Backup Package*

Thanks to the XML-based configuration file used by pfSense® software, backups are a breeze. All of the settings for the system are held in one single file (see *XML Configuration File*). In the vast majority of cases, this one file can be used to restore a system to a fully working state identical to what was running previously. There is no need to make an entire system backup, as the base system files are not modified by a normal, running, system.

Note: In rare cases, packages may store files outside of config.xml, check the package documentation for additional information and backup suggestions.

9.9 Backup Strategies

The optimal backup strategy can be summarized in the following points:

- Take frequent backups
- Keep multiple copies of backups in a safe location off the firewall
- Periodically test backups

The remainder of this section expands on these points.

The best practice is to make a backup after each minor change, and both before and after each major change or series of changes. Typically, an initial backup is taken in case the change being made has undesirable effects. An after-the-fact backup is taken after evaluating the change and ensuring it had the intended outcome. Periodic backups are also helpful, regardless of changes, especially in cases where a manual backup may be missed.

pfSense software makes an internal backup upon each change, and the best practice is to download a manual backup as well. The automatic backups made on each change are useful for reverting to prior configurations after changes have proven detrimental, but are not good for disaster recovery as they are on the system itself and not kept externally. As it is a fairly simple and painless process, administrators should make a habit of downloading a backup now and then and keeping it in a safe place. Backups may be handled easily and automatically using the free **AutoConfigBackup** service.

Tip: Backup files can contain sensitive information, so carefully consider security measures for backups kept off the firewall. If they are on other network file shares, ensure access is restricted. For offline backups, consider physical security measures such as keeping media containing backups in a fire safe and at a remote secure location such as a second office or bank safety deposit box.

If changes have been made to system files, such as custom patches or code alterations, those changes must be backed up manually or with the backup package described in *Backup Files and Directories with the Backup Package*, as they will not be backed up or restored by the built-in backup system. This includes alterations to system files mentioned elsewhere in the documentation, such as `/boot/device.hints`, `/boot/loader.conf.local`, and others.

Note: Custom patches should be handled using the **System Patches** package, which is backed up with config.xml, rather than saving manually patched files.

In addition to making backups, **backups must also be tested**. Before placing a system into production, backup the configuration, wipe the disk, and then attempt some of the different restoration techniques in this chapter. The best practice is to periodically test backups on a non-production machine or virtual machine. The only thing worse than a missing backup is an unusable backup!

RRD graph data can optionally be held in the XML configuration file backup. This behavior is disabled by default due to the resulting size of the backup file. There are also other ways to ensure this data is backed up safely. See *Backup Files and Directories with the Backup Package* later in this chapter.

CHAPTER

TEN

INTERFACE TYPES AND CONFIGURATION

10.1 Interface Configuration

To assign a new interface:

- Navigate to **Interfaces > Assignments**
- Pick the new interface from the **Available network ports** list
- Click  **Add**

The newly assigned interface will be shown in the list. The new interface will have a default name allocated by the firewall such as OPT1 or OPT2, with the number increasing based on its assignment order. The first two interfaces default to the names WAN and LAN but they can be renamed. These OPTx names appear under the **Interfaces** menu, such as **Interfaces > OPT1**. Selecting the menu option for the interface will open the configuration page for that interface.

10.1.1 General Configuration

The following options are available for all interface types.

Description The name of the interface. Interface names may only contain letters, numbers and the only special character that is allowed is an underscore (_).

This changes the name of the interface on the **Interfaces** menu, on the tabs under **Firewall > Rules**, under **Services > DHCP**, and elsewhere throughout the GUI. Using a custom name makes it easier to remember the purpose of an interface and to identify an interface for adding firewall rules or choosing other per-interface functionality.

IPv4 Configuration Type Configures the IPv4 settings for the interface. Details for this option are in the next section, *IPv4 Configuration Types*.

IPv6 Configuration Type Configures the IPv6 settings for the interface. Details for this option are in *IPv6 Configuration Types*.

MAC address The MAC address of an interface can be changed ("spoofed") to mimic a previous piece of equipment, depending on the type of interface.

> **Warning:** The best practice is to not force a specific MAC address. The old MAC address will generally be cleared out by resetting the equipment to which this firewall connects, or by clearing the ARP table, or waiting for the old ARP entries to expire. Changing the MAC address is a long-term solution to a temporary problem.

Spoofing the MAC address of the previous firewall can allow for a smooth transition from an old router to a new router, so that ARP caches on devices and upstream routers are not a concern. It can also be used to fool a piece of equipment into believing that it's talking to the same device that it was talking to before, as in cases where a certain network router is using static ARP or otherwise filters based on MAC address. This is common on cable modems, where they may require the MAC address to be registered if it changes.

Note: ARP cache problems tend to be very temporary, resolving automatically within minutes or by power cycling other equipment.

One downside to spoofing the MAC address is that unless the old piece of equipment is permanently retired, there is a risk of later having a MAC address conflict on the network, which can lead to connectivity problems.

If the old MAC address must be restored, this option must be emptied out and then the firewall *must* be rebooted. Alternately, enter the original MAC address of the network card and save/apply, then empty the value again.

MTU (Maximum Transmission Unit) The Maximum Transmission Unit (**MTU**) size field can typically be left blank, but can be changed when required. Some situations may call for a lower MTU to ensure packets are sized appropriately for an Internet connection. In most cases, the default assumed values for the WAN connection type will work properly. It can be increased for those using jumbo frames on their network.

On a typical Ethernet style network, the default value is `1500`, but the actual value can vary depending on the interface configuration.

MSS (Maximum Segment Size) Similar to the MTU field, the **MSS** field "clamps" the Maximum Segment Size (MSS) of TCP connections to the specified size in order to work around issues with Path MTU Discovery.

Speed and Duplex The default value for link speed and duplex is to let the firewall decide what is best. That option typically defaults to *Autoselect*, which negotiates the best possible speed and duplex settings with the peer, typically a switch.

The speed and duplex setting on an interface must match the device to which it is connected. For example, when the firewall is set to *Autoselect*, the switch must also be configured for *Autoselect*. If the switch or other device has a specific speed and duplex forced, it must be matched by the firewall.

Switch Port Netgate Appliances with an integrated switch have an option on this page which controls the link state for this interface by having it mirror the state of a switch port. In this way, a firewall interface configured as a VLAN which maps to a switch port can be set to follow the status of the physical switch port. Otherwise, since it is a VLAN attached to an internal uplink, the status would always show as up.

Consult the Netgate Product Manuals for more information on switch configuration.

10.1.2 Reserved Networks

Block Private Networks When **Block private networks** is active, the firewall inserts a rule automatically which prevents any RFC 1918 networks (`10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`) and loopback (`127.0.0.0/8`) from communicating on that interface.

This option is typically only desirable on WAN type interfaces to prevent the possibility of privately numbered traffic coming in over a public interface.

Block bogon networks When **Block bogon networks** is active, the firewall will block traffic from a list of unallocated and reserved networks. This list is periodically updated by the firewall automatically.

> **Warning:** This option should only be used on external interfaces (WANs), it is not necessary on local interfaces and it can potentially block required local traffic.

See *Block Bogon Networks* for more details on how this feature works.

10.2 IPv4 Configuration Types

Once an interface has been assigned, in most cases it will require an IP address. For IPv4 connections, the following choices are available in the **IPv4 Configuration Type** selector on an interface page (e.g. **Interfaces > WAN**):

- None
- Static IPv4
- DHCP
- PPP
- PPPoE
- PPTP
- L2TP

Each of these is described in this document.

10.2.1 None

When **IPv4 Configuration Type** is set to *None*, IPv4 is disabled on the interface. This is useful if the interface has no IPv4 connectivity or if the IPv4 address on the interface is being managed in some other way, such as for a VPN or tunnel interface.

10.2.2 Static IPv4

With **Static IPv4**, the interface contains a manually configured IPv4 address. When chosen, three additional fields are available on the interface configuration screen:

IPv4 Address The IPv4 address for the interface (e.g. `192.168.1.1`).

CIDR Subnet Mask The CIDR Subnet Mask determines the size of the subnet to which the **IPv4 Address** belongs. This must match the value used by other hosts in the same subnet.

IPv4 Upstream Gateway An upstream gateway for IPv4 traffic, if any. Selecting a gateway here will cause the firewall to treat this interface as a WAN-type interface for NAT and related functions.

> **Warning:** Do not set a gateway for internal interfaces such as a LAN or DMZ. A gateway should only be selected for externally-connected interfaces such as a WAN or a private site-to-site link.
>
> Gateways may still be used on internal interfaces for the purpose of static routes without selecting an **IPv4 Upstream Gateway** here on the interfaces screen.

The **IPv4 Upstream Gateway** field is pre-populated with existing IPv4 gateways defined under **System > Routing** (*Gateways*).



The **Add a new gateway** button is a shortcut to create a new gateway if one does not already exist. Clicking that button displays a modal form to add the gateway without leaving this page. Fill in the details requested on the new form:

Default Gateway If this is the only WAN or will be a new default WAN, check this box. The default IPv4 and IPv6 gateways work independently of one another. The two need not be on the same interface. Changing the default IPv4 gateway has no effect on the IPv6 gateway, and vice versa.

Gateway Name The name used to refer to the gateway internally, as well as in places like Gateway Groups, quality graphs, and elsewhere.

Gateway IPv4 The IPv4 address of the gateway. This address must be inside the same subnet as the Static IPv4 address when using this form.

Description A bit of text to indicate the purpose of the gateway.

10.2.3 DHCP

When an interface is set to **DHCP**, the operating system will attempt automatic IPv4 configuration of this interface via DHCP. This option also activates several additional fields on the page. Under most circumstances these additional fields may be left blank.

Hostname Some ISPs require the **Hostname** for client identification. The value in the **Hostname** field is sent as the DHCP client identifier and hostname when requesting a DHCP lease.

Alias IPv4 Address This value used as a fixed IPv4 alias address by the DHCP client since a typical IP Alias VIP cannot be used with DHCP. This can be useful for accessing a piece of gear on a separate, statically numbered network outside of the DHCP scope. One example would be for reaching a cable modem management IP address.

Reject Leases From An IPv4 address for a DHCP server that should be ignored. For example, a cable modem that hands out private IP addresses when the cable sync has been lost. Enter the private IP address of the modem here, e.g. `192.168.100.1` and the firewall will never pick up or attempt to use a an IP address supplied by the specified server.

DHCP VLAN Priority Optionally sets a VLAN Priority tag (802.1p) on DHCP client traffic. Should only be enabled when required by an ISP and with the settings they provide.

Advanced Configuration Enables options to control the protocol timing. In the vast majority of cases this must be left unchecked and the options inside unchanged.

Protocol Timing The fields in this area give fine-grained control over the timing used by `dhclient` when managing an address on this interface. These options are almost

always left at their default values. For more details on what each field controls, see the dhclient man page

Presets Has several options for preset protocol timing values. These are useful as a starting point for custom adjustments or for use when the values need to be reset back to default values.

Configuration Override Enables a field to use a custom `dhclient` configuration file. The full path must be given. Using a custom file is rarely needed, but some ISPs require DHCP fields or options that are not supported by the GUI.

10.2.4 PPP Types

The various PPP-based connection types such as PPP, PPPoE, PPTP, and L2TP are all covered in detail at *PPPs*. When one of these types is selected here on the interfaces screen, their basic options can be changed as described. To access the advanced options, follow the link on this page or navigate to **Interfaces > Assignments** on the **PPPs** tab, find the entry, and edit it there.

10.3 IPv6 Configuration Types

Similar to IPv4, the **IPv6 Configuration Type** controls if and how an IPv6 address is assigned to an interface. There are several different ways to configure IPv6 and the exact method depends on the network to which this firewall is connected and how the ISP has deployed IPv6.

Warning: Every ISP is different and large providers can even vary by region.

The ISP determines IPv6 settings for a circuit, and they are the only valid source for that information. As such, this documentation does not include examples for specific providers. Contact the ISP for information about their IPv6 client settings and requirements.

The ISP should provide instructions and specific values for configuring IPv6 on their service. For example, on a circuit with a static IPv6 configuration the ISP should supply the subnet addresses and prefix values for the WAN itself, as well as for routed prefixes. Providers who require DHCPv6 should supply values for settings such as the prefix delegation size, along with any requirements they have for client behavior.

See also:

For more information on IPv6, including a basic introduction, see *IPv6*.

10.3.1 None

When **IPv6 Configuration Type** is set to *None*, IPv6 is disabled on the interface. This is useful if the interface has no IPv6 connectivity or if the IPv6 address on the interface is being managed in some other way, such as for a VPN or tunnel interface.

10.3.2 Static IPv6

The Static IPv6 controls work identically to the Static IPv4 settings. See *Static IPv4* for details.

With **Static IPv6**, the interface contains a manually configured IPv6 address. When chosen, three additional fields are available on the interface configuration screen: **IPv6 Address**, a prefix length selector, and the **IPv6 Upstream Gateway** field.

The default IPv4 and IPv6 gateways work independently of one another. The two need not be on the same interface. Changing the default IPv4 gateway has no effect on the IPv6 gateway, and vice versa.

10.3.3 DHCP6

DHCP6 configures automatic IPv6 configuration of this interface via DHCPv6. DHCPv6 will configure the interface with an IPv6 address, prefix length, DNS servers, etc. but not a gateway. The gateway is obtained via router advertisements, so this interface will be set to accept router advertisements. This is a design choice as part of the IPv6 specification, not a limitation of this implementation. For more information on router advertisements, see *Router Advertisements*.

Several additional fields are available for IPv6 DHCP that do not exist for IPv4 DHCP:

Use IPv4 Connectivity as Parent Interface When set, the IPv6 DHCP request is sent using IPv4 on this interface, rather than using native IPv6. This is only required in special cases when the ISP requires this type of configuration.

Request only an IPv6 Prefix When set, the DHCPv6 client does not request an address for the interface itself, it only requests a delegated prefix.

DHCPv6 Prefix Delegation Size If the ISP supplies a routed IPv6 network via prefix delegation, they will publish the delegation size, which can be selected here. It is typically a value somewhere between *48* and *64*. For more information on how DHCPv6 prefix delegation works, see *DHCP6 Prefix Delegation*.

Note: To use this delegation, another internal interface must be set to an **IPv6 Configuration Type** of *Track Interface* (*Track Interface*) so that it can use the addresses delegated by the upstream DHCPv6 server.

Send IPv6 Prefix Hint When set, the **DHCPv6 Prefix Delegation Size** is sent along with the request to inform the upstream server how large of a delegation is desired by this firewall. If an ISP allows the choice, and the chosen size is within their allowed range, the requested size will be given instead of the default size.

Debug When set, the DHCPv6 client is started in debug mode.

Do not wait for a RA Informs the operating system not to wait for a router advertisement when configuring the interface. This is required by some ISPs.

Do not allow PD/Address release Prevents the operating system from sending a DHCPv6 release message on exit.

Some ISPs will release the allocated address or prefix when a client sends this message. With this option set, the client is more likely to receive the same allocation with subsequent requests.

DHCPv6 VLAN Priority Optionally sets a VLAN Priority tag (802.1p) on DHCPv6 client traffic. Should only be enabled when required by an ISP and with the settings they provide.

Advanced Configuration Enables a wide array of advanced tuning parameters for the DHCPv6 client. These options are rarely used, and when they are required, the values are dictated by the ISP or network administrator. See the dhcp6c.conf man page for details.

Configuration Override Enables a field to use a custom configuration file. The full path must be given. Using a custom file is rarely needed, but some ISPs require DHCP fields or options that are not supported in the pfSense WebGUI.

10.3.4 SLAAC

Stateless address autoconfiguration (*SLAAC*) as the IPv6 type makes the operating system attempt to configure the IPv6 address for the interface from router advertisements (RA) that advertise the prefix and related information.

Note: DNS is not typically provided via RA, so the firewall will still attempt to get DNS servers via DHCPv6 when using SLAAC. The RDNSS extensions to the RA process may allow DNS servers to be obtained from RA in some cases. For more information on router advertisements, see *Router Advertisements*.

This selection has one additional option:

Use IPv4 connectivity as parent interface When set, IPv6 requests are sent over the IPv4 connectivity layer used by this interface (e.g. PPPoE) rather than the parent interface directly. May be required by certain ISPs.

10.3.5 6RD Tunnel

6RD is an IPv6 tunneling technology employed by ISPs to quickly enable IPv6 support for their networks, passing IPv6 traffic inside specially crafted IPv4 packets between and end user router and the ISP relay. It is related to 6to4 but is intended to be used within the ISP network, using the IPv6 addresses from the ISP for client traffic. To use 6RD, the ISP must supply three pieces of information: The 6RD prefix, the 6RD Border Relay, and the 6RD IPv4 Prefix length.

6RD Prefix The 6RD IPv6 prefix assigned by the ISP, such as `2001:db8::/32`.

6RD Border Relay The IPv4 address of the ISP 6RD relay.

6RD IPv4 Prefix Length Controls how much of the end user IPv4 address is encoded inside of the 6RD prefix. This is normally supplied by the ISP. A value of `0` means the entire IPv4 address will be embedded inside the 6RD prefix. This value allows ISPs to effectively route more IPv6 addresses to customers by removing redundant IPv4 information if an ISP allocation is entirely within the same larger subnet.

10.3.6 6to4 Tunnel

Similar to 6RD, *6to4* is another method of tunneling IPv6 traffic inside IPv4. Unlike 6RD, however, 6to4 uses constant prefixes and relays. As such there are no user-adjustable settings for using the *6to4* option. The 6to4 prefix is always `2002::/16`. Any address inside of the `2002::/16` prefix is considered a 6to4 address rather than a native IPv6 address. Also unlike 6RD, a 6to4 tunnel can be terminated anywhere on the Internet, not only at the end user ISP, so the quality of the connection between the user and the 6to4 relay can vary widely.

6to4 tunnels are always terminated at the IPv4 address of `192.88.99.1`. This IPv4 address is anycasted, meaning that although the IPv4 address is the same everywhere, it can be routed regionally toward a node close to the user.

Another deficiency of 6to4 is that it relies upon other routers to relay traffic between the 6to4 network and the remainder of the IPv6 network. There is a possibility that some IPv6 peers may not have connectivity to the 6to4 network, and

thus these would be unreachable by clients connecting to 6to4 relays, and this could also vary depending upon the 6to4 node to which the user is actually connected.

10.3.7 Track Interface

The *Track Interface* choice works in concert with another IPv6 interface using DHCPv6 Prefix Delegation. When a delegation is received from the ISP, this option designates which interface will be assigned the IPv6 addresses delegated by the ISP and in cases where a larger delegation is obtained, which prefix inside the delegation is used.

IPv6 Interface A list of all interfaces on the system currently set for dynamic IPv6 WAN types offering prefix delegation (DHCPv6, PPPoE, 6rd, etc.). Select the interface from the list which will receive the delegated subnet information from the ISP.

IPv6 Prefix ID If the ISP has delegated more than one prefix via DHCPv6, the IPv6 Prefix ID controls which of the delegated `/64` subnets will be used on this interface. This value is specified in hexadecimal.

For example, If a `/60` delegation is supplied by the ISP that means 16 `/64` networks are available, so prefix IDs from `0` through `f` may be used.

For more information on how prefix delegation works, see *DHCP6 Prefix Delegation*.

10.4 Interface Groups

Unlike the other interfaces in this chapter, an **Interface Group** is not a type of interface that can be assigned. Interface groups are used to apply firewall or NAT rules to a set of interfaces on a common tab. If this concept is unfamiliar, consider how the firewall rules for OpenVPN, the PPPoE server, or L2TP server work. There are multiple interfaces in the underlying OS, but the rules for all of them are managed on a single tab for each type.

If many interfaces of a similar function are present on the firewall that need practically identical rules, an interface group may be created to add rules to all of the interfaces at the same time. Interfaces can still have their own individual rules, which are processed after the group rules.

10.4.1 Interface Group Options

When creating or editing an Interface Group, the following options are available:

Group Name The name of the interface group. Has the same restrictions as the name of an interface. The name may only contain upper and lowercase letters, no numbers, spaces, or special characters.

Group Description An optional text description for reference.

Group Members A multi-select list of assigned interfaces on the firewall from which group members can be added. Add interfaces to the group by selecting them with ctrl-click (PC) or cmd-click (MAC).

10.4.2 Creating an Interface Group

To create an interface group:

- Navigate to **Interfaces > Assignments**, **Interface Groups** tab
- Click  **Add** to create a new group
- Fill in the options as described in *Interface Group Options*
- Click **Save**

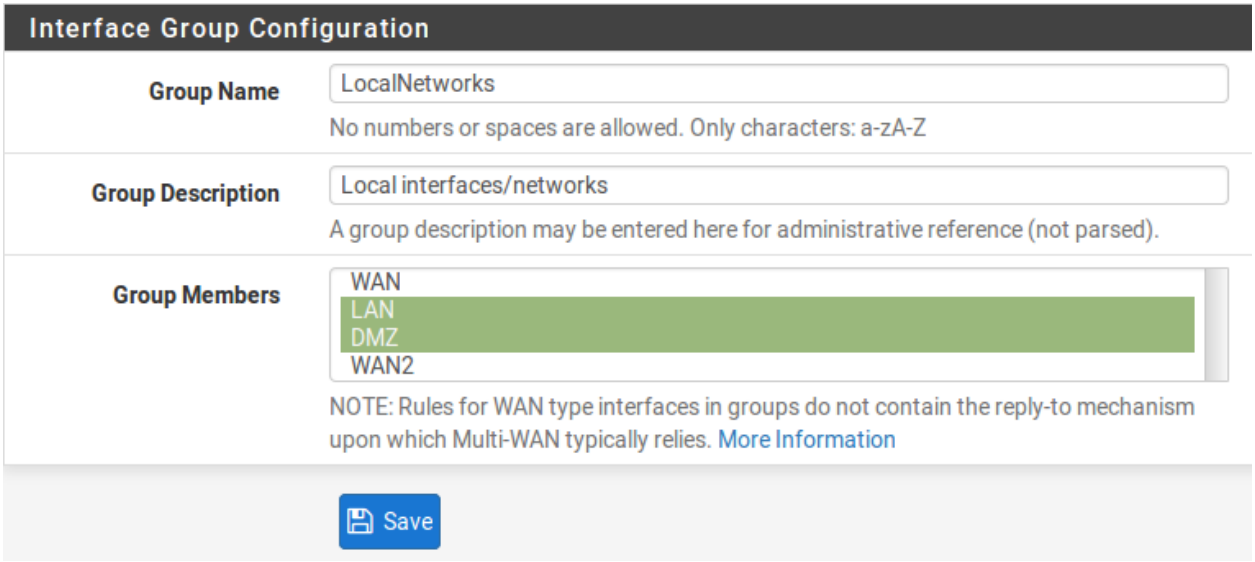


Fig. 1: Add Interface Group

10.4.3 Using an Interface Group

Interface groups each have an individual tab under **Firewall > Rules** to manage their rules. Figure *Interface Group Firewall Rules Tab* shows the firewall rule tab for the group defined in figure *Add Interface Group*

See also:

Configuring firewall rules for information on managing firewall rules.

10.4.4 Group Rule Processing Order

The rule processing order for user rules is:

- Floating rules
- Interface group rules
- Rules on the interface directly

For example, if a rule on the group tab matches a connection, the interface tab rules will not be consulted. Similarly, if a floating rule with **Quick** set matched a connection, the interface group rules will not be consulted.

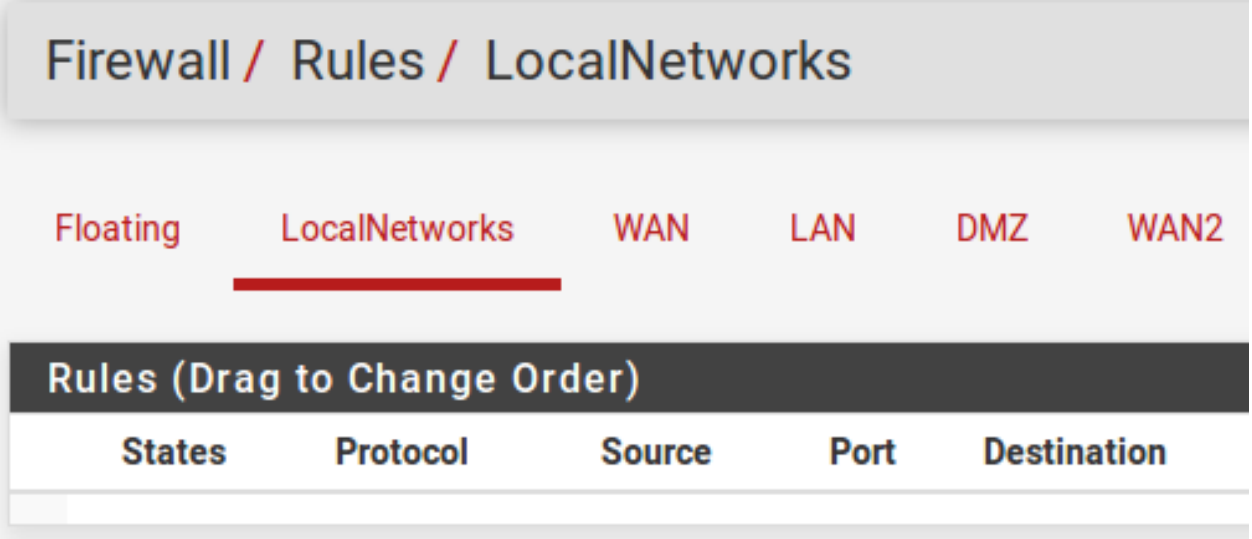


Fig. 2: Interface Group Firewall Rules Tab

The processing order prevents some combination of rules that otherwise might be a good fit. For example, if a general blocking rule is present on the group, it cannot be overriden by a rule on a specific interface. Same with a pass rule, a specific interface rule cannot block traffic passed on a group tab rule.

10.4.5 Use with WAN Interfaces

The best practice is to not use interface groups with multiple WANs. Doing so may appear to be convenient, but the group rules do not receive the same treatment as actual WAN tab rules. For example, rules on a tab for a WAN-type interface will receive `reply-to` which allows pf to return traffic back via the interface from which it entered. Group tab rules do not receive `reply-to` which effectively means that the group rules only function as expected on the WAN with the default gateway.

10.5 PPPs

There are four types of PPP interfaces:

- Plain **PPP** for 3G/4G and modem devices
- **PPPoE** for DSL or similar connections
- **PPTP** and **L2TP** for ISPs that require them for authentication.

In most cases these are managed from the interface settings directly, but they can also be edited under **Interfaces > Assignments** on the **PPPs** tab.

See also:

PPP Logs

10.5.1 Multi-Link PPP (MLPPP)

Multi-Link PPP (MLPPP) can be configured when editing any type of PPP instance by selecting multiple **Link Interface** entries at the same time.

Warning: MLPPP only works on multiple circuits from the same provider where the provider supports MLPPP.

MLPPP bonds multiple PPP links into a single larger aggregate channel. Unlike other multi-WAN techniques, with MLPPP it is possible to use the full bandwidth of all links for a single connection, and the usual concerns about load balancing and failover do not apply. The MLPPP link is presented as one interface with one IP address, and if one link fails, the connection functions the same but with reduced capacity.

For more information on MLPPP, see *Multiple WAN Connections*.

10.5.2 PPP (Point-to-Point Protocol) Interface Types

Add or edit a PPP entry as follows:

- Navigate to **Interfaces > Assignments** on the **PPPs** tab
- Click  to edit an existing entry or  to add a new entry
- Set the **Link Type**

The **Link Type** determines the remaining options on the page. The available link types are explained throughout the remainder of this document.

PPP (3G/4G, Modem)

The **PPP** link type is used for talking to a modem over a serial device. This can be anything from a USB 3G/4G dongle for accessing a cellular network down to an old hardware modem for dial-up access.

Note: Some cellular modems appear as Ethernet devices and not serial devices. Those are configured as regular interfaces, not as PPP devices.

See also:

- *Cellular Wireless*

When configuring a PPP device, the following options are available:

Link Interface A list of serial devices that can be used to communicate with a modem. Click on a specific entry to select it for use.

Note: The serial device for a modem is not automatically detected. Some modems present themselves as several devices, and the subdevice for the PPP line may be any of the available choices, but start with the last device, then try the first, and then others in between if none of those function.

Description A text description of this PPP instance, for reference (e.g. "VZW Modem").

Country The country in which this system resides (e.g. *United States*).

Selecting a **Country** populates the **Provider** list.

Provider The cellular service provider for this modem (e.g. *Verizon*).

Selecting a **Provider** populates the **Plan** list.

Plan The type of cellular service this modem uses from **Provider**.

Selecting a **Plan** populates the remaining fields on the page with known values for that combination of **Provider** and **Plan** in that **Country**.

The remaining options can be configured manually if other values are needed, or when using an unlisted provider:

Username and Password The credentials used for the PPP login, if any.

Phone Number The number to dial at the ISP to gain access. For 3G/4G this tends to be a number such as `*99#` or `#777`, and for dial-up this is usually a traditional telephone phone number.

Access Point Name (APN) This field is required by some ISPs to identify the service to which the client connects. Some providers use this to distinguish between consumer and business plans, or legacy networks.

APN Number Optional setting. Defaults to `1` if the APN is set, and ignored when APN is unset.

SIM PIN Security code on the SIM to prevent unauthorized use of the card. Do not enter anything here if the SIM does not have a PIN.

SIM PIN Wait Number of seconds to wait for SIM to discover network after the PIN is sent to the SIM. If the delay is not long enough, the SIM may not have time to initialize properly after unlocking.

Init String The modem initialization string, if necessary. Do not include `AT` at the beginning of the command. Most modern modems do not require a custom initialization string.

Connection Timeout Time to wait for a connection attempt to succeed, in seconds. Default is 45 seconds.

Uptime Logging When checked, the uptime for the connection is tracked and displayed on **Status > Interfaces**.

PPPoE (Point-to-Point Protocol over Ethernet)

PPPoE is a popular method of authenticating and gaining access to an ISP network, most commonly found on DSL networks, but may also be used on fiber or other link types.

Warning: Due to limitations in the way PPPoE frames are processed by network cards, incoming PPPoE traffic is limited to a single network interface queue. As such, performance may be limited or otherwise lower than expected. See *PPPoE with Multi-Queue NICs* for details.

To configure a PPPoE link, start by setting **Link Type** to *PPPoE* and complete the remainder of the settings as follows:

Link Interface(s) A list network interfaces that can be used for PPPoE. These are typically physical interfaces but it can also work over some other interface types such as VLANs. Select one for normal PPPoE, or multiple for MLPPP.

Description An optional text description of the PPP entry.

Username and Password The credentials for this PPPoE circuit. These will be provided by the ISP, and the username is typically in the form of an e-mail address, such as `mycompany@ispexample.com`.

Service Name Left blank for most ISPs, some require this to be set to a specific value. Contact the ISP to confirm the value if the connection does not function when left blank.

Configure NULL Service Name Some ISPs require `NULL` be sent instead of a blank service name. Check this option when the ISP considers this behavior necessary.

Periodic Reset Configures a pre-set time when the connection will be dropped and restarted. This is rarely needed, but in certain cases it can better handle reconnections when an ISP has forced daily reconnections or similar quirky behavior.

PPTP (Point-to-Point Tunneling Protocol)

Not to be confused with a PPTP VPN, this type of PPTP interface is meant to connect to an ISP and authenticate, much the same as PPPoE works. The options for a PPTP WAN are identical to the PPPoE options of the same name. Refer to the previous section for configuration information.

L2TP (Layer 2 Tunneling Protocol)

L2TP, as it is configured here, is used for connecting to an ISP that requires it for authentication as a type of WAN. L2TP works nearly identically to PPTP. Refer to the previous sections for configuration information.

L2TP has one additional option not found on other types:

Shared Secret A shared secret used to authenticate the tunnel connection and encrypt control L2TP control packets. This **must** match the shared secret set on the L2TP server. May be left blank if the server does not support a shared secret.

10.5.3 Advanced PPP Options

All PPP types have several advanced options in common that can be edited in their entries here. In most cases these settings need not be altered. To show these options, click  **Display Advanced**.

Dial On Demand The default behavior for a PPP link is to immediately connect and immediately attempt to reconnect when a link is lost. This behavior is described as **Always On**. **Dial-on-Demand** delays this connection attempt. When set, the firewall will wait until a packet attempts to leave the via this interface, and then it will connect. Once connected, it will not automatically disconnect.

Idle Timeout A PPP connection will be held open indefinitely by default. A value in **Idle Timeout**, specified in seconds, will cause the firewall to monitor the line for activity. If there is no traffic on the link for the given amount of time, the link will be disconnected. If **Dial-on-Demand** has also been set, the firewall will return to dial-on-demand mode.

Note: The firewall performs gateway monitoring by default which generates two ICMP pings per second on the interface. **Idle Timeout** will not function in this case. This can be worked around by editing the gateway for this PPP link, and checking **Disable Gateway Monitoring**.

Compression (vjcomp) This option controls whether or not Van Jacobson TCP header compression will be used by this connection. By default it will be negotiated with the peer during login, so if both sides support the feature it will be used. Checking **Disable vjcomp** will cause the feature to always be disabled. Normally this feature is beneficial because it saves several bytes per TCP data packet, when possible. The option should almost always remain enabled.

Note: This compression is ineffective for TCP connections with enabled modern extensions like time stamping or SACK, which modify TCP options between sequential packets.

TCP MSS Fix The tcpmssfix option causes the PPP daemon to adjust incoming and outgoing TCP SYN segments so that the requested maximum segment size (MSS) is not greater than the amount allowed by the interface MTU. This is necessary in most cases to avoid problems caused by routers that drop ICMP "Datagram Too Big" messages. Without these messages, the originating machine sends data, it passes the rogue router then hits a machine that has an MTU that is not big enough for the data. Because the IP "Don't Fragment" option is set, this machine sends an ICMP "Datagram Too Big" message back to the originator and drops the packet. The rogue router drops the ICMP message and the originator never gets to discover that it must reduce the fragment size or drop the IP Don't Fragment option from its outgoing data. If this behavior is undesirable, check **Disable tcpmssfix**.

Note: The MTU and MSS values for the interface may also be adjusted on the interface's configuration page under the **Interfaces** menu, such as **Interfaces > WAN** (*Interface Configuration*).

Short Sequence (ShortSeq) This option is only meaningful if MLPPP is negotiated with the provider. It proscribes shorter multi-link fragment headers, saving two bytes on every frame. It is not necessary to disable this for connections that are not multi-link. If MLPPP is active and this feature must be disabled, check **Disable shortseq**.

Address Control Field Compression (ACFComp) This option only applies to asynchronous link types. It saves two bytes per frame. To disable this, check **Disable ACF Compression**.

Protocol Field Compression (ProtoComp) This option saves one byte per frame for most frames. To disable this, check **Disable Protocol Compression**.

PPPoE has two additional advanced options:

Multilink over single link When set, the firewall will use LCP multi-link extensions over a single link. This ignores the MTU/MRU settings. Only enable if supported by the ISP.

Force MTU When set, overrides the MTU negotiated with the ISP with a higher value known to work on the link.

Warning: This option violates RFC 1661 and can break connectivity. While it may result in faster speed as larger packets can be transferred, there is no guarantee that it will function in the future if the provider makes changes.

10.6 GRE (Generic Routing Encapsulation)

Generic Routing Encapsulation (GRE) is a method of tunneling traffic between two endpoints without encryption. It can be used to route packets between two locations that are not directly connected, which do not require encryption. It can also be combined with a method of encryption that does not perform its own tunneling.

Note: The GRE protocol was originally designed by Cisco, and it is the default tunneling mode on many of their devices.

GRE tunnels can carry either IPv4, IPv6, or both types of traffic at the same time.

10.6.1 GRE Interface Settings

Parent interface The interface upon which the GRE tunnel will terminate. Often this will be WAN or a WAN-type connection.

Remote Address The address of the remote peer. This is the address where the GRE packets will be sent by this firewall; The routable external address at the other end of the tunnel.

Local IPv4/IPv6 Tunnel Address The **internal** IPv4 and IPv6 address for the end of the tunnel on this firewall. The firewall will use this address for its own traffic in the tunnel, and tunneled remote traffic would be sent to this address by the remote peer.

Remote IPv4/IPv6 Tunnel Address The IPv4 and IPv6 address used by the firewall **inside** the tunnel to reach the far side. Traffic destined for the other end of the tunnel must use this address as a gateway for routing purposes.

IPv4/IPv6 Tunnel Subnet The subnet mask for the GRE interface address.

Add Static Route When set, the firewall adds an explicit static route for the remote inner tunnel address/subnet via the local tunnel address. This can help with reaching the remote subnet in cases where other route table entries may select the wrong path to that destination.

Description A short description of this GRE tunnel for documentation purposes.

10.6.2 GRE Interface Management

To create or manage a GRE interface:

- Navigate to **Interfaces > Assignments**, **GRE** tab

Note: The items in this list are managed in the usual way. See *Managing Lists in the GUI*.

- Click  **Add** to create a new GRE instance
- Complete the settings as described in *GRE Interface Settings*
- Click **Save**
- Navigate to **Interfaces > Assignments**
- Select the new GRE interface in the **Available network ports** list
- Click  **Add**
- Note the name given to the new interface (e.g. OPT1)
- Navigate to **Interfaces > <name>** where `<name>` corresponds to the name of the GRE interface (e.g. OPT1)
- Check **Enable interface**
- Enter a new name for the interface in **Description** (optional)
- Click **Save**

Then use the interface as any other WAN-type interface. The firewall automatically creates a dynamic gateway for routing purposes. Depending on the use case, the interface may need NAT or firewall rules, static routes, and so on.

10.7 GIF (Generic tunnel InterFace)

A Generic Tunneling Interface (GIF) is similar to *GRE*; Both protocols are a means to tunnel traffic between two hosts without encryption. In addition to tunneling IPv4 or IPv6 directly, GIF may be used to tunnel IPv6 over IPv4 networks and vice versa. GIF tunnels are commonly used to obtain IPv6 connectivity to a tunnel broker such as Hurricane Electric in locations where IPv6 connectivity is unavailable.

See also:

See *Configuring IPv6 Through A Tunnel Broker Service* for information about connecting to a tunnel broker service.

GIF interfaces carry more information across the tunnel than can be done with GRE, but GIF is not as widely supported. For example, a GIF tunnel is capable of bridging layer 2 between two locations while GRE cannot.

GIF interfaces can carry IPv4 or IPv6 traffic, but not both at the same time.

Note: Support for GIF varies by vendor, but is not as common as others like GRE.

10.7.1 GIF Interface Settings

Parent interface The interface upon which the GIF tunnel will terminate. Often this will be WAN or a WAN-type connection.

GIF Remote Address The address of the remote peer. This is the address where the GIF packets will be sent by this firewall; The routable external address at the other end of the tunnel. For example, in a IPv6-in-IPv4 tunnel to Hurricane Electric, this would be the **IPv4 address** of the tunnel server, such as `209.51.181.2`.

GIF tunnel local address The **internal address** for the end of the tunnel on this firewall. The firewall will use this address for its own traffic in the tunnel, and tunneled remote traffic would be sent to this address by the remote peer. For example, when tunneling IPv6-in-IPv4 via Hurricane Electric, they refer to this as the **Client IPv6 Address**.

GIF tunnel remote address The address used by the firewall **inside** the tunnel to reach the far side. Traffic destined for the other end of the tunnel must use this address as a gateway for routing purposes. For example, when tunneling IPv6-in-IPv4 via Hurricane Electric, they refer to this as the **Server IPv6 Address**.

GIF Tunnel Subnet The subnet mask or prefix length for the interface address. Typically `64`.

ECN Friendly Behavior The ECN friendly behavior option controls whether or not the Explicit Congestion Notification (ECN)-friendly practice of copying the TOS bit into/out of the tunnel traffic is performed by the firewall. By default the firewall clears the TOS bit on the packets or sets it to `0`, depending on the direction of the traffic. With this option set, the bit is copied as needed between the inner and outer packets to be more friendly with intermediate routers that can perform traffic shaping. This behavior breaks RFC 2893 so it must only be used when both peers agree to enable the option.

Outer Source Filtering When set, the firewall will not automatic filter based on the outer GIF source. This is normally desirable as it ensures a match with the configured remote peer, which is more secure. When disabled, martian and inbound filtering is not performed which allows asymmetric routing of the outer traffic. This is less secure, but some GIF peers may source traffic in this manner.

Description A short description of this GIF tunnel for documentation purposes.

10.7.2 GIF Interface Configuration

To create or manage a GIF interface:

- Navigate to **Interfaces > Assignments**, **GIF** tab

  Note: The items in this list are managed in the usual way. See *Managing Lists in the GUI*.

- Click  **Add** to create a new GIF instance
- Complete the settings as described in *GIF Interface Settings*
- Click **Save**
- Navigate to **Interfaces > Assignments**
- Select the new GIF interface in the **Available network ports** list
- Click  **Add**
- Note the name given to the new interface (e.g. OPT1)
- Navigate to **Interfaces > <name>** where `<name>` corresponds to the name of the GIF interface (e.g. OPT1)
- Check **Enable interface**
- Enter a new name for the interface in **Description** (optional)
- Click **Save**

Then use the interface as any other WAN-type interface. The firewall automatically creates a dynamic gateway for routing purposes. Depending on the use case, the interface may need NAT or firewall rules, static routes, and so on.

10.8 LAGG (Link Aggregation)

Link aggregation is handled by `lagg(4)` type interfaces (LAGG) on pfSense® software. LAGG combines multiple physical interfaces together as one logical interface. There are several ways this can work, either for gaining extra bandwidth, redundancy, or some combination of the two.

Note: LACP will only work across multiple switches if the switches are Stackable.

10.8.1 LAGG Interface Settings

When creating or editing a LAGG interface, the following settings are available:

Parent Interfaces This list contains all currently unassigned interfaces, plus members of the current LAGG interface when editing an existing instance.

To add interfaces to this LAGG, select one or more interfaces in this list.

Note: An interface may only be added to a LAGG group if it is not assigned. If an interface is not present in the list, it is likely already assigned as an interface.

LAGG Protocol The operating modes for LAGG interfaces are: LACP, Failover, Load Balance, Round Robin, and None.

LACP The most commonly used LAGG protocol. This mode supports IEEE 802.3ad Link Aggregation Control Protocol (LACP) and the Marker Protocol. In LACP mode, negotiation is performed with the switch – which must also support LACP – to form a group of ports that are all active at the same time. This is knowns as a Link Aggregation Group, or LAG. The speed and MTU of each port in a LAG must be identical and the ports must also run at full- duplex. If link is lost to a port on the LAG, the LAG continues to function but at reduced capacity. In this way, an LACP LAGG bundle can gain both redundancy and increased bandwidth.

Traffic is balanced between all ports on the LAG, however, for communication between two single hosts it will only use one single port at a time because the client will only talk to one MAC address at a time. For multiple connections through multiple devices, this limitation effectively becomes irrelevant. The limitation is also not relevant for failover.

In addition to configuring this option on the firewall, the switch must enable LACP on these ports or have the ports bundled into a LAG group. Both sides must agree on the configuration in order for it to work properly.

LACP Timeout Mode controls how often the firewall sends LACP PDUs. An LACP timeout occurs when three consecutive PDUs are missed.

Slow Default. LACP PDUs are sent every `30` seconds. A timeout occurs after `90` seconds.

Fast LACP PDUs are sent every second. A timeout occurs after `3` seconds.

Failover When using the Failover LAGG protocol traffic will only be sent on the *primary* interface of the group. If the primary interface fails, then traffic will use the next available interface.

Note: By default, traffic may only be received by the active interface. Create a system tunable for `net.link.lagg.failover_rx_all` with a value of `1` to allow traffic to be received on every member interface.

Failover mode has one additional option:

Failover Primary Interface This option sets the primary interface for failover mode, or **auto** to allow the firewall to select the primary interface automatically. In auto mode, the first selected interface in the list is primary.

Each non-primary interface is eligible for use in failover if the primary fails.

Load Balance Load Balance mode accepts inbound traffic on any port of the LAGG group and balances outgoing traffic on any active ports in the LAGG group. It is a static setup that does not monitor the link state nor does it negotiate with the switch. Outbound traffic is load balanced based on all active ports in the LAGG using a hash computed using several factors, such as the source and destination IP address, MAC address, and VLAN tag.

Round Robin This mode accepts inbound traffic on any port of the LAGG group and sends outbound traffic using a round robin scheduling algorithm. Typically this means that traffic will be sent out in sequence, using each interface in the group in turn.

None This mode disables traffic on the LAGG interface without disabling the interface itself. The OS will still believe the interface is up and usable, but no traffic will be sent

or received on the group.

Description A short note about the purpose of this LAGG instance.

10.8.2 LAGG Interface Configuration

To create or manage LAGG interfaces:

- Navigate to **Interfaces > Assignments**, **LAGGs** tab
- Click  **Add** to create a new LAGG, or click  to edit an existing instance.
- Complete the settings as described in *LAGG Interface Settings*
- Click **Save**

After creating a LAGG interface, it works like any other physical interface. Assign the lagg interface under **Interfaces > Assignments** and give it an IP address, or build other things on top of it such as VLANs.

Note: If the only purpose of the LAGG interface is to carry VLANs, it does not need to be assigned.

10.8.3 LAGG and Traffic Shaping

Due to limitations in FreeBSD, `lagg(4)` does not support `altq(4)` so it is not possible to use the traffic shaper on LAGG interfaces directly. `vlan(4)` interfaces support `altq(4)` and VLANs can be used on top of LAGG interfaces, so using VLANs can work around the problem. As an alternate workaround, Limiters can control bandwidth usage on LAGG interfaces.

10.8.4 LAGG Throughput

Using a LAGG does not necessarily guarantee full throughput equal to the sum of all interfaces. In particular, a single flow will not exceed the throughput of a LAGG member interface. Traffic on a LAGG is hashed in such a way that flows between two hosts, such as this firewall and an upstream gateway, would only use a single link since the flow is between a single MAC address on each side.

In networks where many hosts communicate with different MAC addresses, the usage can approach the sum of all interfaces in the LAGG.

10.9 QinQ Configuration

QinQ, also known as IEEE 802.1ad or stacked VLANs, is a means of nesting VLAN tagged traffic inside of packets that are already VLAN tagged, or "double tagging" the traffic.

See also:

- *Virtual LANs (VLANs)*

QinQ is used to move groups of VLANs over a single link containing one outer tag, as can be found on some links between locations from ISPs or datacenters. QinQ can be a quick and easy way of trunking VLANs across locations without having a trunking-capable connection between the sites, provided the infrastructure between the locations does not strip tags from the packets.

10.9.1 QinQ Interface Settings

When creating or editing a QinQ interface entry, the following options are available:

Parent Interface The interface that will carry the QinQ traffic.

First level tag The outer VLAN ID on the QinQ interface, or the VLAN ID given by the provider for the site-to-site link.

Adds interface to QinQ interface groups When checked, a new interface group will be created called **QinQ** that can be used to filter all of the QinQ subinterfaces at once.

When hundreds or potentially thousands of QinQ tags are present, this greatly reduces the amount of work needed to use the QinQ interfaces

Description Optional text for reference, used to identify the entry

Member(s) Member VLAN IDs for QinQ tagging. These can be entered one per row or in ranges such as `100-150`.

Click  **Add Tag** to add another line for more tags or ranges.

10.9.2 QinQ Interface Configuration

Setting up QinQ interfaces is fairly simple:

- Navigate to **Interfaces > Assignments**
- Click the **QinQ** tab
- Click  **Add** to add a new QinQ entry
- Configure the QinQ entry as described in *QinQ Interface Settings*
- Click **Save** to complete the interface

10.9.3 QinQ Example

In the following example (Figure *QinQ Basic Example*), a QinQ interface is configured to carry tagged traffic for VLANs `10` and `20` across the link on `igb3` with a first level tag of `2000`.

In Figure *QinQ List*, this entry is shown on the QinQ tab summary list.

The automatic interface group, shown in Figure *QinQ Interface Group*, must not be manually edited. Because these interfaces are not assigned, it is not possible to make alterations to the group without breaking it. To re-create the group, delete it from this list and then edit and save the QinQ instance again to add it back.

Rules may be added to the **QinQ** tab under **Firewall > Rules** to pass traffic in both directions across the QinQ links.

From here, how the QinQ interfaces are used is mostly up to the needs of the network. Most likely, the resulting interfaces may be assigned and then configured in some way, or bridged to their local equivalent VLANs (e.g. bridge an assigned `igb2.10` to `igb3.2000.10` and so on).

The QinQ configuration will be roughly the same on both ends of the setup. For example, if both sides use identical interface configurations, then traffic that leaves Site A out on `igb3.2000.10` will go through VLAN `2000` on `igb3`, come out the other side on VLAN `2000` on `igb3` at Site B, and then in `igb3.2000.10` at Site B.

See also:

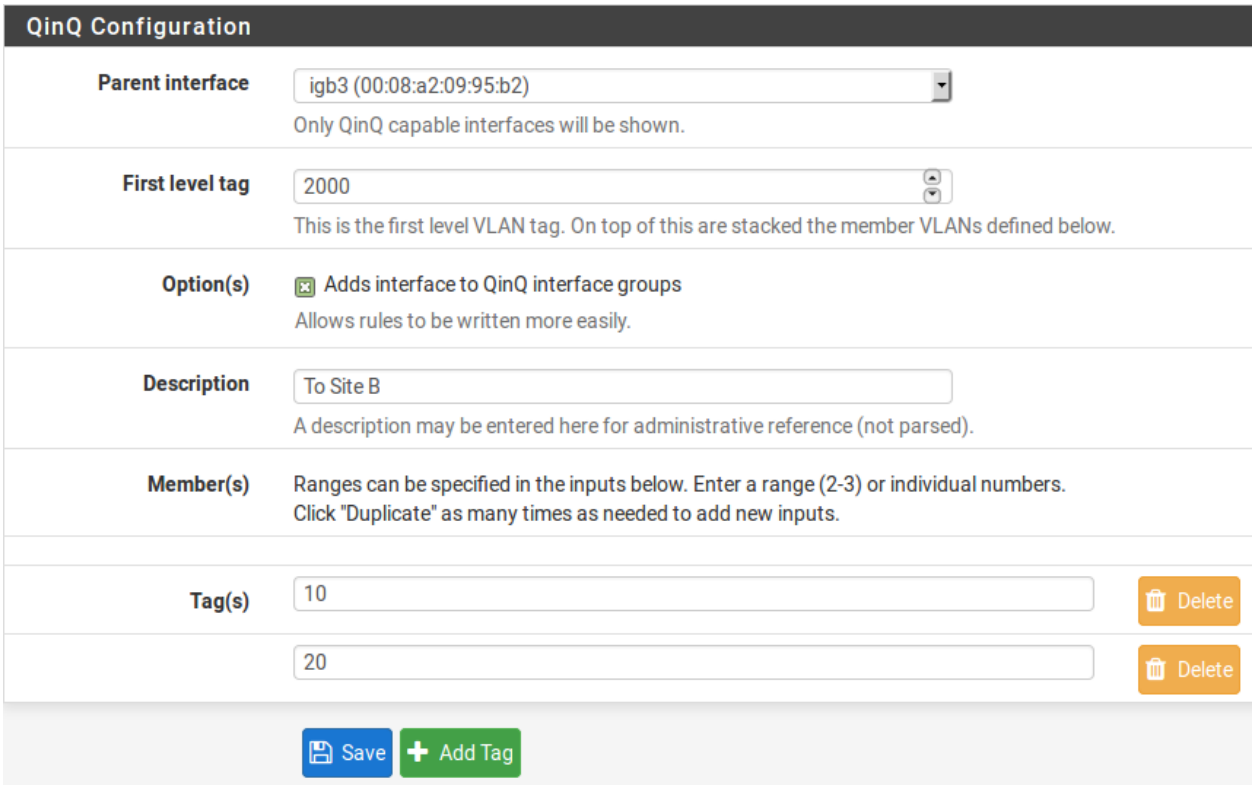


Fig. 3: QinQ Basic Example

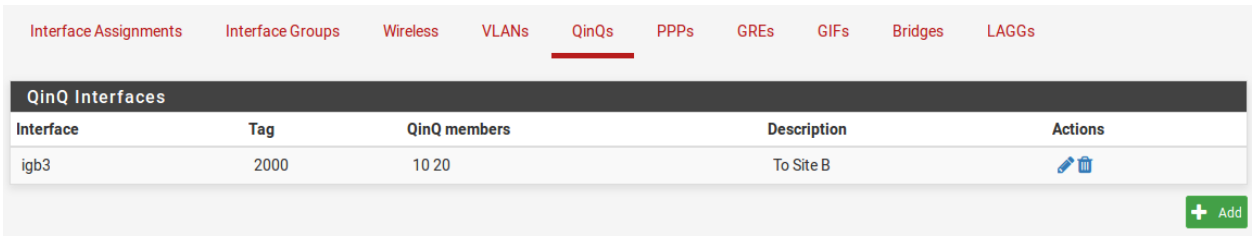


Fig. 4: QinQ List

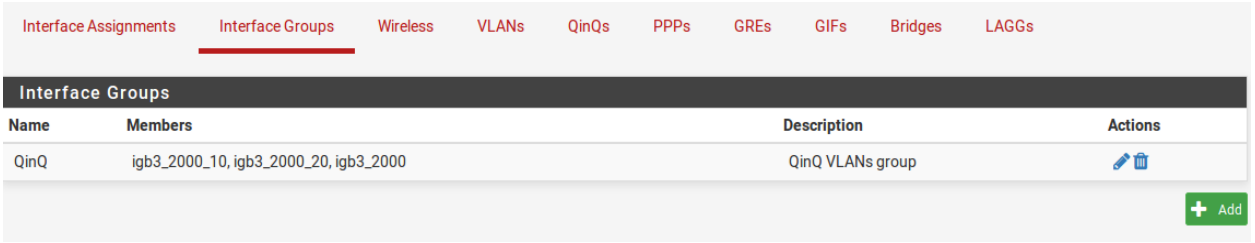


Fig. 5: QinQ Interface Group

- *Virtual LANs (VLANs)*
- *Bridging*
- *Wireless*

pfSense® software supports numerous types of network interfaces, either using physical interfaces directly or by employing other protocols such as PPP or VLANs.

Interface assignments and the creation of new virtual interfaces are all handled under **Interfaces > Assignments**.

10.10 Physical and Virtual Interfaces

Most interfaces discussed in this chapter can be assigned as WAN, LAN, or an OPT interface under **Interfaces > Assignments**. All currently-defined and detected interfaces are listed directly on **Interfaces > Assignments** or in the list of interfaces available for assignment. By default, this list includes only the physical interfaces, but the other tabs under **Interfaces > Assignments** can create virtual interfaces which can then be assigned.

Interfaces support various combinations of options. They can also support multiple networks and protocols on a single interface, or multiple interfaces can be bound together into a larger capacity or redundant virtual interface.

All interfaces are treated equally; Every interface can be configured for any type of connectivity or role. The default WAN and LAN interfaces can be renamed and used in other ways.

Physical interfaces and virtual interfaces are treated the same once assigned, and have the same capabilities. For example, a VLAN interface can have the same type of configuration that a physical interface can have. Some interface types receive special handling once assigned, which are covered in their respective sections of this chapter.

This section covers the various types of interfaces that can be created, assigned, and managed.

10.11 Switches

Some Netgate Appliances sold in the Netgate Store contain built-in switches which can be configured in the GUI under **Interfaces > Switches**. Documentation for the switch configuration can vary by model, and may be found in the Netgate Product Manuals which match a given product.

10.12 Limitations

While the firewall does not impose any limits on the number of interfaces, large numbers of interfaces may function in suboptimal ways. For example, the firewall may take much longer configure interfaces, and the GUI may have rendering issues with large numbers of tabs or menu entries.

Most hardware will accommodate as many physical interfaces as can fit into the case. Issues may vary from driver to driver but generally are hardware-related and not the result of the operating system or pfSense software.

Note: With a large number of physical interfaces, the number of mbufs will likely need to be increased. See *Tuning and Troubleshooting Network Cards*.

Physical limitations aside, significant numbers of virtual interfaces such as VLANs, LAGGs, VPNs, and more may be added to the firewall. These types interfaces tend to outnumber physical interfaces, especially VLANs.

Issues reported by users with large numbers of interfaces (physical and virtual) vary by hardware, configuration, and browser. These issues tend to increase as the number of interfaces approaches 200. Should a particular environment

require more than 128 interfaces, consider alternate designs that do not involve using all of the interfaces on the firewall directly. If the firewall must handle large numbers of interfaces, be wary of potential performance and GUI concerns.

CHAPTER

ELEVEN

USER MANAGEMENT AND AUTHENTICATION

11.1 Default Username and Password

The default credentials for a pfSense® software installation are:

Username `admin`

Password `pfsense`

11.2 Privileges

Managing privileges for users and groups is done similarly, so both will be covered here rather than duplicating the effort. Whether a user or group is managed, the entry must be created and saved first before privileges can be added to the account or group.

To add privileges, edit an existing user or group and click  **Add** in the **Assigned Privileges** or **Effective Privileges** section.

The GUI presents a list of all available privileges. Privileges may be added one at a time by selecting a single entry, or by multi-select using ctrl-click or cmd-click. If other privileges are already present on the user or group, they are hidden from this list so they cannot be added twice. To search for a specific privilege by name, enter the search term in the **Filter** box and click  **Filter**.

Selecting a privilege will show a short description of its purpose in the information block area under the permission list and action buttons. Most of the privileges are self-explanatory based on their names, but a few notable permissions are:

WebCfg - All Pages Grants the user access to any page in the GUI

WebCfg - Dashboard (all) Grants the user access to the dashboard page and all of its associated functions (widgets, graphs, etc.)

WebCfg - System: User Password Manager Page If the user has access to only this page, they can login to the GUI to set their own password but do nothing else.

User - VPN - IPsec xauth Dialin Allows the user to connect and authenticate for IPsec xauth

User - Config - Deny Config Write Prevents the user from making changes to the firewall configuration (`config.xml`).

> **Warning:** This does not prevent the user from taking other actions that do not involve writing to the configuration.

User - System - Shell account access Grants the user the ability to login over SSH, though the user will not have root-level access so functionality is limited. A package for *sudo* is available to enhance this feature.

After login, the firewall will attempt to display the dashboard. If the user does not have access to the dashboard, the GUI will forward the user to the first page in their privilege list to which they have access.

Menus on the firewall only contain entries for which privileges exist on a user account. For example, if the only Diagnostics page that a user has access to is **Diagnostics > Ping** then no other items will be displayed in the **Diagnostics** menu.

11.3 Manage Local Users

The **Users** tab under **System > User Manager** is where individual users are managed.

Note: The admin user cannot be deleted and its username may not be changed.

11.3.1 Creating and Editing Users

The first step is always to add the user and save. Privileges can only be added to existing users, they cannot be added when creating a new user.

Tip: If multiple users need the same privileges, the most efficient method is to add a *group* and then add users to the group.

To add a new user:

- Navigate to **System > User Manager**
- Click  **Add**

To edit an existing user:

- Navigate to **System > User Manager**
- Click  on the row containing the user

11.3.2 User Settings

When creating or editing a user, the following options are available:

Disabled This checkbox controls whether this user will be active. To deactivate this account, enable the option.

Username Sets the login name for the user. This field is required, must be 16 characters or less and may only contain letters, numbers, and a period, hyphen, or underscore.

Password / Confirm Password The password for this user. Ensure the two fields match to confirm the password.

Note: Passwords are stored in the configuration as hashes, not plain text.

Full Name Optional field which can be used to enter a longer name or a description for this user account.

Expiration Date Optional date at which the firewall will automatically deactivate this user account. The date must be entered in `MM/DD/YYYY` format.

Custom Settings Enables options for per-user custom GUI settings. See *Per-user GUI Options and Dashboard Layout* for details.

Group Memberships If one or more groups exist on the firewall (*Manage Local Groups*), this control can add the user as a member.

To add a group for this user:

- Click the group name in the **Not Member Of** column
- Click  to move it to the **Member Of** column

To remove a group from the user:

- Click the group name in the the **Member Of** column
- Click  to move it to the **Not Member Of** column

Effective Privileges A list of privileges this user has, either directly assigned or inherited by group membership.

Appears only when editing an existing user, not when creating a user.

Privilges assigned to the user may be edited by these controls, but group privileges cannot. Group privileges must be managed on the *group*.

See also:

See *Privileges* for information on managing privileges.

Certificate Certificates associated with this user account.

The behavior of this section changes depending on whether the page is creating a new user or editing an existing user. This section is disabled if there are no internal certificate authorities defined on the firewall capable of signing a certificate.

To create a certificate while adding a user:

- Check **Click to create a user certificate**
- Fill in the **Descriptive name**

- Choose a **Certificate Authority**
- Select a **Key Type** and **Key Length**
- Select a **Digest Algorithm**
- Enter a **Lifetime**

See also:

For more information on these parameters, see *Create an Internal Certificate*.

When editing a user, this section of the page instead becomes a list of certificates associated with this user account.

To create a certificate for an existing user:

- Click click  **Add**
- Fill in the settings on the page as described in *Create an Internal Certificate* (some data is pre-filled)

To associate an existing certificate with this user:

- Set **Method** to *Choose an Existing Certificate*
- Select an entry from the **Existing Certificate** list
- Click **Save**

Authorized SSH keys Public keys for SSH and SCP authentication.

To add a key, paste or enter in the key data. Multiple keys are allowed, one per line.

> **Warning:** Only enter authorized keys into this field. Do not add them to files in user home directories. Those files will be overwritten by the GUI the next time account information is synchronized to disk (e.g. at boot time).

IPsec Pre-Shared Key Pre-Shared Key (PSK) for for this user to connect to a non-xauth Pre-Shared Key mobile IPsec setup.

If a PSK is entered here, the username is used as the identifier. The PSK is also displayed under **VPN > IPsec** on the **Pre-Shared Keys** tab.

Note: This field has no effect for IKEv2 or xauth mobile IPsec.

Per-user GUI Options and Dashboard Layout

Each user can have their own settings for various GUI options and their dashboard layout. To enable this for a user, check the **Custom Settings** box when adding or editing the user.

When that option is active, additional GUI options for the user are present on the user account page. Additionally, the user can have their own personal dashboard layout, starting from the system-wide layout.

Choose the other GUI options desired for the user such as theme, top navigation, host name in menu, dashboard columns, show/hide associated panels, left column labels and browser tab text.

Tip: Users with the **WebCfg - System: User Settings** privilege may adjust their own GUI options.

Users in the `admin` group already have this privilege.

A user with **Custom Settings** enabled and the **User Settings** privilege will have menu option **System > User Settings**. The user can select this to change the GUI options for their account.

When a user with **Custom Settings** adds, moves or removes dashboard widgets, the custom dashboard layout is saved in the preferences for only that user.

11.4 Manage Local Groups

Groups manage sets of user privileges so they do not need to be maintained individually on every user account. For example, a group can be used for IPsec xauth users, or a group that can access the firewall dashboard, a group of firewall administrators, or many other possible scenarios using any combination of privileges.

Groups are managed under **System > User Manager** on the **Groups** tab.

Note: The `all` and `admins` groups cannot be deleted.

11.4.1 Groups and Remote Authentication

When working with group privileges while authenticating against LDAP and RADIUS (*Authentication Servers*), local groups must exist with names that exactly match groups from the server. For example, if an LDAP group named `firewall_admins` exists then the firewall must also contain a identically named group, `firewall_admins`, with the desired privileges.

If a user attempts to authenticate against a remote authentication server and there are no matching groups, the user will not have any privileges from groups, and cannot access resources which require privileges.

11.4.2 Creating and Editing Groups

As with users, the first step is to add the group and save. Privileges can only be added to existing groups, they cannot be added when creating a new group.

To add a new group:

- Navigate to **System > User Manager**, **Groups** tab
- Click  **Add**

To edit an existing group:

- Navigate to **System > User Manager**, **Groups** tab
- Click  on the row containing the group

11.4.3 Group Settings

Group name The name of the group.

For groups in the *Local* scope, this setting has the same restrictions as a username: It must be 16 characters or less and may only contain letters, numbers, and a period, hyphen, or underscore.

Groups in the *Remote* scope do not have strict name restrictions, for example they may have longer names.

Scope The scope in which this group is available for use.

Note: LDAP and RADIUS groups can match names in both local and remote scopes.

Local Groups on the firewall itself, such as those for use in the shell, filesystem, and other local uses. These groups are added to the operating system, so they are subject to naming restrictions imposed there.

Remote Groups from remote sources, such as authentication servers (RADIUS or LDAP). These groups are not exposed to the operating system, and thus are only available for use in the GUI and other similar uses not involving the operating system layer. This scope has relaxed name restrictions, for example, group names may be longer and may contain spaces.

Description Optional free-form text for reference and to better identify the purpose of the group in case the **Group name** is not sufficient.

Group Memberships This set of controls defines which existing users will be members of the new group. Firewall users are listed in the **Not Members** column by default.

To add a user to this group:

- Click the user name in the **Not Members** column
- Click  to move it to the **Members** column

To remove a user from this group:

- Click the user name in the the **Members** column
- Click  to move it to the **Not Members** column

Assigned Privileges A list of privileges assigned to this group. Appears only when editing an existing group.

See also:

See *Privileges* earlier in this for information on managing privileges.

11.5 Authentication Servers

The firewall can use RADIUS and LDAP servers to authenticate users from remote sources.

User Manager Support contains information on which areas of the firewall support these servers

To add a new server:

- Navigate to **System > User Manager**, **Authentication Servers** tab
- Click  **Add**

To edit an existing server, click  next to its entry on the same page.

Each type of authentication server is covered in the following documents

11.5.1 RADIUS Authentication Servers

Remote Authentication Dial-In User Service (RADIUS) is a protocol commonly supported by a wide variety of networking equipment for user authentication, authorization, and accounting (AAA).

Servers are commonly available as well, including *FreeRADIUS* and *Active Directory via NPS*.

Though most areas on pfSense® software which support RADIUS now integrate their RADIUS settings via the user manager, a few remain which use separate settings, such as the PPPoE and L2TP servers.

See also:

- *Controlling Client Parameters via RADIUS*

> **Warning:** Secure the link between the firewall and the RADIUS server. If the server is local, use a trusted management network. If the server is remote, communicate only over VPN tunnels.
>
> Some RADIUS protocols transmit passwords in plain text, and though others attempt to protect the password in other ways, other aspects of the protocol are not encrypted and may contain sensitive information.

RADIUS Configuration

Descriptive name The name for this RADIUS server. This name will be used to identify the server throughout the GUI.

Protocol The protocol used by the firewall when performing RADIUS requests. May be one of:

PAP Password Authentication Protocol. Sends passwords unencrypted, and is considered weak. It is more widely supported than other methods, and may be required by specific features (e.g. mOTP).

> **Warning:** Due to its security deficiencies, avoid using PAP where possible.

MD5-CHAP Challenge-Handshake Authentication Protocol using MD5 hashing. The RADIUS server sends a challenge value and the client responds with a hash of the challenge value and the password together. More secure than PAP as it does not transmit passwords in the clear, but both parties must know the plain text of the password.

MS-CHAPv1 A Microsoft variation of CHAP where neither side needs to know the plain text of the password. Though it is generally more secure, it has other known weaknesses which make it vulnerable to attack.

MS-CHAPv2 An updated variation of MS-CHAPv1. It is used in EAP as well as 802.1x/WPA Enterprise for wireless. However, it also has known weaknesses.

Note: Certain RADIUS features may require specific modes. For example, mOTP typically requires PAP since it reads the password in the clear to separate the PIN and OTP code. Services utilizing EAP typically use MS-CHAPv2.

Hostname or IP address The address of the RADIUS server. This can be a fully qualified domain name or an IPv4 IP address.

> **Warning:** The RADIUS client on the firewall does not currently support IPv6.

Shared Secret The password established for this firewall *on the RADIUS server* software.

Services offered This selector sets which services are offered by this RADIUS server.

Authentication The firewall will use this RADIUS server to authenticate users.

Accounting The firewall will send RADIUS start/stop accounting packet data for login sessions if supported in the area where it is used.

Authentication and Accounting The server will be used for both types of actions.

Authentication port Only appears if an Authentication mode is chosen. Sets the UDP port where RADIUS authentication will occur. The default RADIUS authentication port is `1812`.

Accounting port Only appears if an Accounting mode is chosen. Sets the UDP port where RADIUS accounting will occur. The default RADIUS accounting port is `1813`.

Authentication Timeout Controls how long, in seconds, that the RADIUS server may take to respond to an authentication request. If left blank, the default value is `5` seconds. If an interactive two-factor authentication system is in use, increase this timeout to account for how long it will take the user to receive and enter a token, which can be 60-120 seconds or more if it must wait for an external action such as a phone call, SMS message, etc.

RADIUS NAS IP Attribute Sets the value the firewall will send in the RADIUS request `NAS-IP-Address` attribute. This value is used by the RADIUS server to identify this firewall. The server can use this value to make authentication decisions, or to denote which node users were authenticated by in accounting data.

In most cases, the `NAS-IP-Address` value does not matter so long as it is unique to this firewall. However, more complicated RADIUS environments may use this attribute to let the server make more informed decisions about users logging into different services. For example, if there are multiple Captive Portal instances on the firewall, multiple RADIUS server entries can be created, each using the specific interface address for a given portal. The RADIUS server could then choose to only let certain sets of users login to each portal.

Adding a RADIUS Server

To add a new RADIUS server:

- Add the firewall as a client on the RADIUS server
- Navigate to **System > User Manager**, **Authentication Servers** tab
- Click  Add
- Set the **Type** selector to *RADIUS*

  The GUI will change the form to display RADIUS Server Settings
- Fill in the fields as described in *RADIUS Configuration*
- Click **Save** to create the server
- Navigate to **Diagnostics > Authentication** to test the RADIUS server using a valid account.

RADIUS Groups

There are two requirements for RADIUS groups to function properly:

- The RADIUS server must return a list of groups in the `Class` RADIUS reply attribute as a string.
- The same groups must exist locally (*Manage Local Groups*)

Multiple groups groups returned by the RADIUS server in the `Class` attribute must be separated by a semicolon. For example, in FreeRADIUS, to return the `admins` and `VPNUsers` groups, use the following Reply-Item RADIUS Attribute:

```
Class := "admins;VPNUsers"
```

If the RADIUS server returns the group list properly for a user, and the groups exist locally, then the groups will be listed on the results when using the **Diagnostics > Authentication** page to test an account.

If the groups do not show up when testing, ensure the groups exist in the *Group Manager* with matching names and that the server is returning the `Class` attribute as a string, not binary.

11.5.2 LDAP Authentication Servers

Though Lightweight Directory Access Protocol (LDAP) is technically a repository for user information, it also supports mechanisms for user authentication via bind operations.

There are many popular user directory implementations which use LDAP, including Active Directory, OpenLDAP, FreeIPA, and more.

Note: LDAP server implementations and schemas vary widely. As such, there are no complete and specific examples in this document.

LDAP Configuration

Hostname or IP address The address of the LDAP server. This can be a fully qualified domain name, an IPv4 IP address, or an IPv6 IP address.

Note: If this LDAP server uses SSL, the value of this field **must** match the certificate presented by the LDAP server. Typically this means it must be a hostname which resolves to the IP address of the LDAP server, but the specific requirements depend on the contents of the server certificate.

For example, with a value of `ldap.example.com` in this field, the server certificate must include an FQDN value of `ldap.example.com`, and `ldap.example.com` must resolve to `192.168.1.5`. One exception to this is if the IP address of the server also happens to be the listed in the server certificate.

This can be worked around in some cases by creating a DNS host override to make the server certificate hostname resolve to the correct IP address if they do not match in this network infrastructure and they cannot be easily fixed.

Port value This setting specifies the port on which the LDAP server is listening for LDAP queries. The default port is `389` for Standard TCP and STARTTLS, and `636` for SSL. This field is updated automatically with the proper default value based on the selected **Transport**.

Note: When using port `636` for SSL, the firewall uses an `ldaps://` URL, not STARTTLS. Ensure that the LDAP server is listening on the correct port with the correct mode.

Transport This setting controls which transport method will be used by the firewall to communicate with the LDAP server.

Warning: LDAP queries will contain sensitive data, such as usernames, passwords, and other information about the user. The best practice is for the firewall to use encryption when communicating with the LDAP server, if the LDAP server supports it. Both SSL/TLS and STARTTLS will encrypt traffic between the firewall and the LDAP server.

Standard TCP (Default) Plain unencrypted TCP connections on port `389`. This is not secure, but is widely supported and also useful for debugging with packet captures. Do not use this protocol across untrusted networks.

STARTTLS Encrypted Connects using TCP port `389` but negotiates encryption with the server using STARTTLS.

Note: Not all LDAP servers support STARTTLS, check the LDAP server documentation and configuration.

SSL/TLS Encrypted Connects using SSL/TLS on TCP port `636` to encrypt LDAP queries.

Note: Not all LDAP servers support SSL/TLS, check the LDAP server documentation and configuration.

Peer Certificate Authority The CA chosen with this selector is used by the firewall to validate the LDAP server certificate when **Transport** is set to **SSL/TLS Encrypted** or **STARTTLS Encrypted** mode.

The selected CA must match the CA which signed the LDAP server certificate, otherwise validation will fail. If the LDAP server is using a globally trusted certificate (e.g. Let's Encrypt or another public CA), choose *Global Root CA List*.

See *Certificate Authority Management* for more information on creating or importing CAs.

Client Certificate (Factory only) This certificate is sent to the LDAP server to identify this client when using an encrypted transport mode. If the LDAP server requires a client certificate, the server will use this certificate to ensure that the firewall is authorized to make LDAP queries.

This certificate must be issued by the CA used by the LDAP server to validate connecting clients.

Protocol version Chooses which version of the LDAP protocol is employed by the LDAP server, either *2* or *3*, typically *3*.

Server Timeout The time, in seconds, after which LDAP operations are considered as failed. Using a lower value will allow the GUI to try other authentication sources faster when the server fails. If the LDAP server is slow or overloaded, a larger value can help the firewall accept delayed responses.

Search scope Determines where, and how deep, an LDAP search will be performed to locate a match.

Level Controls the depth of the LDAP search.

One Level Search only one level, defined by the **Authentication Containers**.

Entire Subtree Search the entire subtree of the directory, starting with the **Authentication Containers**.

Tip: This is typically the best choice, and is nearly always required for Active Directory configurations.

Base DN Controls where the search will start. Typically set to the root of the LDAP structure, e.g. `DC=example,DC=com`

Authentication containers A list of potential account locations or containers, separated by semicolons. These containers will be prepended to the **Base DN** above when the firewall crafts LDAP queries. Alternately, specify a full container path here and leave the **Base DN** blank.

Tip: If the LDAP server supports it, and the bind settings are correct, click  **Select a container** to browse the LDAP server and select containers from a list.

Some examples of containers are:

- `CN=Users;DC=example;DC=com` This searches for users inside of the domain component `example.com`, a common syntax for Active Directory
- `CN=Users,DC=example,DC=com;OU=OtherUsers,DC=example,DC=com` This searches in two different locations, the second of which is restricted to the `OtherUsers` organizational unit.

Extended Query Specifies an extra restriction to query after the username, which allows group membership to be used as a filter. This must include both the item to search as well as the method of searching. For example, a restriction based on group membership would use `memberOf`. Check the LDAP server documentation for information on forming such queries.

To set an extended query, check the box and fill in the **Query** value with a filter such as:

```
memberOf=CN=VPNUsers,CN=Users,DC=example,DC=com
```

Bind credentials Controls how this LDAP client will attempt to bind to the server.

Note: Active Directory typically requires the use of bind credentials and may need a service account or administrator-equivalent depending on the server configuration. Consult Windows documentation to determine which is necessary in a specific environment.

Bind Anonymous (Default) When checked the firewall will use anonymous binds. When unchecked the GUI presents the **Bind Credentials** fields.

Bind Credentials (User DN/Password) When **Bind Anonymous** is unchecked, the credentials in these fields are used by the firewall to make authenticated binds when performing a query.

The **User DN** may be a username or a full DN, depending on what the LDAP server requires.

Attributes

Initial Template This option only appears when initially creating an LDAP server entry. It pre-fills the remaining options on the page with common defaults for a given type of LDAP server. The choices include *OpenLDAP*, *Microsoft AD*, and *Novell eDirectory*.

User naming attribute The attribute used to identify the name of a user, most commonly `cn` or `samAccountName`.

Group naming attribute The attribute used to identify a group, such as `cn`.

Group member attribute The attribute of a user that signifies it is the member of a group, such as `member`, `memberUid`, `memberOf`, or `uniqueMember`.

RFC2307 Groups Specifies how group membership is organized on the LDAP server. When unset (default), the queries assume the server uses Active Directory style group membership (RFC 2307bis) where groups are listed as an attribute of the user object. When checked, queries use RFC 2307 style group membership where the users are listed as members on the group object.

Note: In this mode the `Group member attribute` will typically be set to `memberUid`, but may vary by LDAP schema.

RFC2307 User DN When set, queries include the user DN when searching for groups.

Group Object Class Specifies the object class of RFC 2307 style groups. Typically `posixGroup` but it may vary by LDAP schema. Not necessary for Active Directory style groups.

Shell Authentication Group DN The LDAP group DN for users allowed to login via SSH. This is used with the **Shell Authentication** option on the *Settings* tab to allow LDAP users to login via SSH.

To login via SSH, users must be a member of this group and have valid `posixAccount` attributes in their LDAP account.

UTF8 Encode When checked, queries to the LDAP server are encoded for UTF-8 and the responses are decoded from UTF-8. Support varies depending on the LDAP server. Generally only necessary if user names, groups, passwords, and other attributes contain UTF-8 or international style accented characters.

Username Alterations When unchecked, a username given as `user@hostname` will have the `@hostname` portion stripped so only the username is sent in the LDAP bind request. When checked, the username is sent in full.

Allow Unauthenticated Bind When set, bind requests with empty passwords will be rejected locally. Some LDAP servers, specifically Microsoft Active Directory, will accept unauthenticated bind requests and treat them as successful.

> **Warning:** This behavior must be disabled on the LDAP server where possible. Allowing requests to succeed with an empty password is a significant security risk and it affects any device or service authenticating against an LDAP server.
>
> Though this option allows the firewall to reject such authentication attempts, other LDAP clients may not offer the same choice. Disabling the feature on the server is the most secure means of correcting the problem. Consult the LDAP server documentation for information on disabling this behavior.

Adding an LDAP Server

To add a new LDAP server:

- Make sure that the LDAP server can be reached by the firewall
- Import the Certificate Authority used by the LDAP server before proceeding if using SSL/TLS or STARTTLS encryption

  See *Certificate Authority Management* for more information on creating or importing CAs.
- Navigate to **System > User Manager**, **Authentication Servers** tab
- Click  **Add**
- Set the **Type** selector to *LDAP*

  The GUI will change the form to display LDAP server settings
- Fill in the fields as described previously in *LDAP Configuration*
- Click **Save** to create the server
- Visit **Diagnostics > Authentication** to test the LDAP server using a valid account

LDAP Groups

There are two requirements for LDAP groups to function properly:

- The LDAP authentication settings must match the group membership style used by the LDAP server
- The same groups must exist locally (*Manage Local Groups*)

If the LDAP query returns the group list properly for a user, and the groups exist locally, then the groups will be listed on the results when using the **Diagnostics > Authentication** page to test an account.

If the groups do not show up, ensure they exist in the *Group Manager* with matching names and that the proper group structure is present on the LDAP authentication server entry (e.g. RFC 2703 options.)

See also:

- pfSense Hangouts on Youtube to view the August 2015 Hangout on RADIUS and LDAP.

- *External User Authentication Examples*

11.6 Settings

The **Settings** tab in the User Manager controls how the firewall authenticates users for the GUI and SSH.

Session Timeout This field specifies how long a GUI login session will last when *idle*. This value is specified in **minutes**, and the default is four hours (`240` minutes). A value of `0` may be entered to disable session expiration, making the login sessions valid forever. A shorter timeout is better, though it should be long enough that an active administrator would not be logged out unintentionally while making changes.

> **Warning:** Allowing a session to stay valid when idle for long periods of time is insecure. If an administrator leaves a terminal unattended with a browser window open and logged in, someone or something else could take advantage of the open session.

Authentication Server This selector chooses the primary authentication source for users logging into the GUI. This can be a RADIUS or LDAP server, or the default *Local Database*.

Note: If the RADIUS or LDAP server is unreachable, the authentication will fall back to *Local Database* even if another method is chosen.

Shell Authentication When set, the selected **Authentication Server** will also be configured as the authentication source for SSH access to the firewall. By default, only accounts in the *User Manager* with shell privileges can login over SSH.

This works with both RADIUS and LDAP servers, with some caveats:

RADIUS Servers When used with a RADIUS server, accounts must exist on the firewall with the same names and the expected privileges. They will authenticate against RADIUS but use the local accounts settings otherwise.

LDAP Servers When used with an LDAP server, the **Shell Authentication Group DN** must be set on the *LDAP Authentication Server* entry. Users must be a member of that group and have valid `posixAccount` attributes in their LDAP account.

Auth Refresh Time Time in seconds for which the firewall cache authentication results. The default is `30` seconds, maximum `3600` (one hour). Shorter times result in more frequent queries to authentication servers.

The firewall periodically re-authenticates users against the remote server to ensure the account is still valid and has the expected privileges. Checking frequently is more secure, but puts a larger burden on the authentication server and can increase page load times on the firewall.

11.6.1 Remote Authentication Servers and Privileges

When using a RADIUS or LDAP server to authentication for the GUI, the *users* and/or *group memberships* must be defined in the firewall in order to properly allocate permissions, as there is no method to obtain permissions dynamically from an authentication server.

For group membership to work properly, the firewall must be able to recognize the groups as presented by the authentication server. This requires two things:

- The local groups must exist with identical names (*Manage Local Groups*).
- The firewall must be able to locate or receive a list of groups from the authentication server.

See *Authentication Servers* for details specific to each type of authentication server.

11.7 Logging Out of the GUI

To end a GUI login session navigate to **System > Logout** or close the browser window.

Sessions will automatically expire if they are idle for longer than the **Session Timeout** defined on **System > User Manager**, **Settings** tab. The default session timeout is 4 hours (240 minutes) of idle time.

See also:

- *Sudo Package*
- *External User Authentication Examples*
- *Granting Users Access to SSH*
- *Accessing the Firewall Filesystem with SCP*
- *Authenticating Users with Google Cloud Identity*
- *Troubleshooting Authentication*
- *Troubleshooting Access when Locked Out of the Firewall*

The User Manager in pfSense® software provides the ability to create and manage multiple user accounts. These accounts can be used to access the GUI, use VPN services like IPsec and OpenVPN, and use the Captive Portal.

The User Manager is located at **System > User Manager**. From there users, groups, servers may be managed, and settings that govern the behavior of the User Manager may be changed.

The User Manager can also be used to define external authentication sources such as RADIUS and LDAP.

See also:

pfSense Hangouts on Youtube to view the February 2015 Hangout on User Management and Privileges, and the August 2015 Hangout on RADIUS and LDAP.

11.8 User Manager Support

As of this writing, not all areas of the firewall hook back into the User Manager.

GUI Supports users in the User Manager, and via RADIUS or LDAP. Groups or Users from RADIUS or LDAP require definitions in the local User Manager to manage their access permissions.

SSH/SCP Supports users from the User Manager, and via RADIUS or LDAP. Requires special privilege granted to users or groups.

IPsec Supports users in the User Manager, RADIUS or LDAP via User Manager for Xauth, and RADIUS for IKEv2 with EAP-RADIUS.

OpenVPN Supports users in the User Manager, RADIUS or LDAP via User Manager.

Captive Portal Support local users, RADIUS, or LDAP via User Manager.

L2TP Supports users in the L2TP settings, and via RADIUS in the L2TP settings.

PPPoE Server Supports users in the PPPoE settings, and via RADIUS in the PPPoE settings.

CHAPTER

TWELVE

CERTIFICATE MANAGEMENT

12.1 Certificate Properties

Certificate authority and certificate entries have several properties in common. The common properties of both types are covered here.

12.1.1 Keys

The public and private keys of the certificate are used for cryptographic operations.

Key Type Certificate key type can be either **RSA** or **ECDSA** (Elliptic Curve Digital Signature Algorithm).

RSA RSA keys are more common and well-supported than ECDSA, as well as having some performance benefits.

Key Length When using RSA keys, the security is proportional to the key size. Larger keys are more secure, but they also take longer to generate and are slower to use. RSA performance decreases rapidly as the key size increases.

The best practice is to not use keys smaller than `2048` bits where possible. Legacy and embedded systems may not support larger keys.

ECDSA ECDSA is a newer method, and is not as widely adopted. Its main advantage is that is can use smaller keys to provide equivalent levels of security to RSA. ECDSA is slower at verifying signatures than RSA, but scales better.

Curve Name There are a variety of ECDSA curves available, but only a few have been confirmed to work with various services on the firewall. The services which support each curve are noted in the list. Pick the curve based on which services will use this certificate authority or certificate.

12.1.2 Digest Algorithm

Digest Algorithms, also known as Message Digest Algorithms and Hash Algorithms, are used to create a fixed-length hash of content for signing.

The larger the hash, the stronger it is and the less likely it is to be susceptible to collisions which compromise the integrity of the hash. The current best practice is to use a minimum of SHA-256.

Warning: Though the GUI still contains support for SHA-1, it is considered weak and should not be used. Rare exceptions can be made for legacy systems which do not support stronger hashes.

12.1.3 Lifetime

The Lifetime of a certificate authority or certificate determines the length, in days, for which the certificate is valid. Shorter lifetimes are more secure, but require more work as the certificates must be renewed or replaced more frequently.

See also:

Renew or Reissue a CA or Certificate.

For certificate authorities, a longer lifetime such as `3650` days (10 years) is acceptable.

Certificates for users typically also have a long lifetime, but specific values depend largely on the needs of an organization. The GUI defaults to `3650` days for User Certificates, but it a better practice is to use a lower value when practical.

Server certificates have stricter requirements for their lifetime. The current accepted maximum lifetime for server certificates is `398` days. Most browsers and other software will no longer accept new server certificates with longer lifetimes.

Note: Another special case is server certificates obtained using *ACME from Let's Encrypt*. These only have a lifetime of `90` days, but since they are automatically replaced well before they expire, there is little extra administrative overhead once the initial setup is complete.

12.1.4 Distinguished Name

The entity to which a certificate authority or certificate belongs, also known as the Subject, is identified by the unique components of the certificate. The primary component for this purpose is the Distinguished Name (DN). These are typically filled in with an organization's information, or in the case of an individual, personal information. This information is mostly cosmetic, and used to verify the accuracy of the CA, and to distinguish one CA from another.

A DN is composed of several fields which contain information about the subject.

Only the **Common Name** is required, the other fields may be left blank.

Warning: A DN with less unique information has the potential to be misidentified later when comparing certificate subjects. Always fill in enough information to uniquely identify the subject.

Common Name A short name, such as a username or hostname. Do not use spaces or punctuation, other than that which is typically found in a hostname.

Note: This name is not used directly for certificate validation on modern systems, which look at *Subject Alternative Name* values instead.

Country Code The two letter ISO country code for the certificate subject location.

Note: The ISO country code is not the same as a the hostname TLD code for a country.

State or Province The geographical state or province name for the certificate subject location. This value should be spelled out, not using an abbreviation or code.

City The city for the certificate subject location.

Organization The name of the organization to which the subject belongs. For example, a company name, government agency name, or similar.

Organizational Unit A division or department inside the organization, if any. For example, "IT Department" or "Accounting".

Note: When creating a certificate, the GUI populates most of these fields with the values from the certificate authority chosen for signing. The contents of the fields may be changed before performing the signing operation.

12.1.5 Subject Alternative Name

The Subject Alternative Name (SAN) list is only present on certificates. It contains information used to validate the identity of the certificate. For example, when connecting to a device on the network, a system may compare the hostname or IP address to which it connected with values in the certificate SAN list. This way, it can be sure it is communicating with the intended host and not an impostor.

Note: The **Common Name** value from a certificate is automatically added to the SAN list internally, as its inclusion is a requirement of current standards.

The following types of SAN entries can be added to a certificate:

FQDN or Hostname A fully qualified domain name (e.g. `host.domain.tld`) or a hostname (`host`). In most cases this hostname would also exist in DNS. In the case of user certificates, this could also be a username.

IP Address An IP address (e.g. `x.x.x.x`), typically an address found on a network device using this certificate. Necessary for clients to properly validate the certificate when connecting by IP address instead of by hostname.

URI A Uniform Resource Identifier for the certificate subject. In practice, only used as an alternate way to determine the hostname when communicating with servers. It does not restrict certificate validity to specific URIs on a server.

E-mail Address An e-mail address for the certificate subject.

12.1.6 Certificate Properties in Lists

When viewing the lists of CA and certificate entries, the properties of the entry are available in the **Distinguished Name** column. The DN is printed there and additional detailed information is available from the  icon.

Underneath that information, the GUI prints the start and end dates for the validity of the entry. The difference between the start and end date is the **Lifetime**. When an entry is nearing expiration, the GUI highlights the end date in yellow. When an entry is expired, it is red. The system also generates notifications for expiring certificates.

See also:

The certificate expiration warning threshold is `27` days by default, but can be customized. See *Notifications* for details.

12.2 Certificate Authority Management

Certificate Authority (CA) entries are managed from **System > Cert Manager**, on the **CAs** tab.

See also:

Renew or Reissue a CA or Certificate

12.2.1 Certificate Authority Settings

When creating a CA entry, the following options are available:

Trust Store Controls whether or not this CA is added to the certificate trust store on the firewall. When added to the trust store, a CA will be considered valid for all certificate operations performed by the operating system. If the firewall must contact a server using a certificate issued by a private CA, this allows such certificates to be trusted by client programs such as LDAP authentication, SMTP notifications, URL table connections, and many others.

Randomize Serial Controls whether or not the CA will randomize serial numbers when it signs certificates or if it will use a sequential serial number.

The current best practice is to randomize serial numbers so they are unpredictable. This also reduces the chances of generating two certificates with the same serial number in circumstances where the CA is moved between different hosts or signs certificates in multiple places.

Common Properties See *Certificate Properties* which covers the remaining fields on the page.

When importing or editing an existing CA entry, the following options are available:

Certificate Data The PEM-encoded certificate data for the CA.

Certificate data is typically contained in a file ending with `.crt` or `.pem`. It would be plain text, and enclosed in a block such as:

```
-----BEGIN CERTIFICATE-----
[A bunch of random-looking base64-encoded data]
-----END CERTIFICATE-----
```

The format varies slightly for ECDSA certificates.

Certificate Private Key The PEM-encoded private key for the CA. If this is omitted, the CA cannot sign certificates or CRLs, but it can be used for other purposes. When empty, the CA is marked as "External". They key can be filled in later to enable signing and to have the CA treated as "Internal".

The key data is typically in a file ending in `.key`. It would be plain text data enclosed in a block such as:

```
-----BEGIN RSA PRIVATE KEY-----
[A bunch of random-looking base64-encoded data]
-----END RSA PRIVATE KEY-----
```

The format varies slightly for ECDSA keys.

Next Certificate Serial The serial number of the next certificate, used when the CA is not set to randomize serial numbers.

It is essential that each certificate have a unique serial, or there will be problems later with certificate revocation. If the next serial is unknown, attempt to estimate how many certificates have been made from the CA, and then set the number high enough a collision would be unlikely.

12.2.2 Create a new Certificate Authority Entry

To create a new CA entry, start the process as follows:

- Navigate to **System > Cert Manager**, **CAs** tab
- Click **Add** to create a new a CA
- Enter a **Descriptive name** for the CA

  This is used as a label for this CA throughout the GUI.
- Select the **Method** that best suits how the CA will be generated

  Create an Internal Certificate Authority Creates a new root CA. Fill in the settings as described in *Certificate Authority Settings*.

  Import an Existing Certificate Authority Exports a CA certificate created on another host, with or without a private key. This can be useful in two ways: One, for CAs made using another system, and two, for CAs made by others that must be trusted.

  Fill in the settings as described in *Certificate Authority Settings*.

  Note: If the CA has been signed by an intermediary and not directly by a root CA, then import each entry in the chain separately, starting with the root CA.

  Create an Intermediate Certificate Authority Creates a new intermediate CA, to be signed by another internal CA on this firewall.

  Pick an existing internal CA for the **Signing Certificate Authority** and fill in the remaining settings as described in *Certificate Authority Settings*.

If errors are reported, such as invalid characters or other input problems, they will be described on the screen. Correct the errors, and attempt to Save again.

12.2.3 Edit a Certificate Authority

To edit an existing CA:

- Navigate to **System > Cert Manager**, **CAs** tab
- Locate the CA entry in the list
- Click the  icon at the end of its row

The edit screen presented by the GUI allows editing the fields as if the CA were being imported.

For information on the fields on this screen, see *Certificate Authority Settings*. In most cases the purpose of this screen would be to add the CA to the trust store, correct the **Serial** of the CA if needed, or to add a key to an imported CA so it can be used to create and sign certificates and CRLs.

12.2.4 Export a Certificate Authority

To export a CA:

- Navigate to **System > Cert Manager**, **CAs** tab
- Locate the CA entry in the list
- Click the  icon at the end of its row to export the CA certificate.

  The file will download with the descriptive name of the CA as the file name, with the extension `.crt`.
- Click the  icon to export the private key for the CA if necessary

  The file will download with the descriptive name of the CA as the file name, with the extension `.key`.

  In most cases the private key for a CA would not be exported unless the CA is being moved to a new location or a backup is being made. When using the CA for a VPN or most other purposes, only export the certificate for the CA and do not export the key.

  Warning: If the private key for a CA gets into the wrong hands, the other party could generate new certificates that would be considered valid against the CA.

12.2.5 Remove a Certificate Authority

To remove a CA, first it must be removed from active use.

- Check areas that can use a CA, such as OpenVPN, IPsec, and packages.

  Note: In most cases, the areas using a CA are noted in the **In Use** column of the CA list. This does not necessarily include all areas, especially if the CA is used by a package.

- Remove entries utilizing the CA or select a different CA
- Navigate to **System > Cert Manager**, **CAs** tab
- Locate the CA entry in the list

- Click  at the end of the row for the CA
- Click OK on the confirmation dialog

12.2.6 Renew a Certificate Authority

To renew a CA entry:

- Navigate to **System > Cert Manager**, **CAs** tab
- Locate the CA entry in the list
- Click  at the end of the row for the CA
- Follow the rest of the renewal procedure as described in *Renew or Reissue a CA or Certificate*

12.3 Certificate Management

Certificates are managed from **System > Cert Manager**, on the **Certificates** tab.

When creating a certificate on any platform the process generally follows this flow:

- User creates a certificate signing request (CSR) and set of keys. The public key is a part of the CSR, but the private key is separate.
- The user transmits only the CSR to the CA, not the private key which remains private to the user.
- The CA signs the CSR, which results in a certificate.
- The CA transmits the certificate to the user.

The user now has a certificate trusted by the CA, and the private key for the certificate.

The GUI handles most this process automatically, but it also supports performing individual steps separately as well. For example, when creating an internal certificate, there is no need to create and sign a CSR in separate steps, the GUI automates that process and does them in one step. Aside from that, the GUI supports creating a CSR which can be sent to a separate CA and it also supports signing CSRs.

12.3.1 Certificate Settings

When creating a certificate entry or working with a CSR, the following common options are available:

Common Properties See *Certificate Properties* which covers properties of most certificate entries.

Certificate Type Sets the intended purpose of this certificate. This influences which key usage properties are set in the certificate and thus limits the ways in which the certificate can operate.

> **Warning:** The certificate can only be used for purposes which match the selected type. Attempting to use it in other ways will produce errors and fail, or prevent the certificate from being shown for selection.

User Certificate Certificates for end users and clients. For example, IPsec and OpenVPN client certificates.

> **Note:** User type certificates include Extended Key Usage attributes indicating they may be used for client authentication. They also are marked with a constraint indicating that they are not a CA.

Server Certificate Certificates for servers, services, daemons, etc. For example, HTTPS servers (GUI, Captive Portal, HAProxy, etc), IPsec IKEv2 mobile server, OpenVPN servers, and for packages such as FreeRADIUS.

> **Note:** Server type certificates include Extended Key Usage attributes indicating they may be used for server authentication as well as the OID `1.3.6.1.5.5.8.2.2` which is used by Microsoft to signifiy that a certificate may be used as an IKE intermediate. These are required for Windows 7 and later to trust the server certificate for use with certain types of VPNs. They also are marked with constraints indicating that they are not a CA, and they have `nsCertType` set to `server`.

Alternative Names Identifiers for this certificate, such as a hostname. See *Subject Alternative Name* for details.

When importing an existing certificate entry, the following options are available:

Certificate Data The PEM-encoded certificate data for the certificate.

Certificate data is typically contained in a file ending with `.crt` or `.pem`. It would be plain text, and enclosed in a block such as:

```
-----BEGIN CERTIFICATE-----
[A bunch of random-looking base64-encoded data]
-----END CERTIFICATE-----
```

The format varies slightly for ECDSA certificates.

Private Key Data The PEM-encoded private key for the certificate.

The key data is typically in a file ending in `.key`. It would be plain text data enclosed in a block such as:

```
-----BEGIN RSA PRIVATE KEY-----
[A bunch of random-looking base64-encoded data]
-----END RSA PRIVATE KEY-----
```

The format varies slightly for ECDSA keys.

12.3.2 Create a new Certificate

To create a new certificate, start the process as follows:

- Navigate to **System > Cert Manager**, **Certificates** tab
- Click **Add** to create a new certificate
- Enter a **Descriptive name** for the certificate

  This is used as a label for this certificate throughout the GUI.
- Select the **Method** that best suits how the certificate will be generated

  These options and further instructions are in the corresponding sections below:

  - Create an Internal Certificate
  - Import an Existing Certificate
  - Create a Certificate Signing Request
  - Sign a Certificate Signing Request
- Complete the steps for the chosen method
- Click Save to finish the import process

Create an Internal Certificate

The most common **Method** is *Create an Internal Certificate*. This will make a new certificate using one of the existing certificate authorities.

- Select the **Certificate Authority** which will sign this certificate. Only a CA that has a private key present can be in this list, as the private key is required in order for the CA to sign a certificate.
- Set the properties of the certificate as described in *Certificate Settings*.
- Click **Save**.

Import an Existing Certificate

To import an existing certificate from an external source, set **Method** to *Import an Existing Certificate*. This can be useful for certificates made using another system or for certificates provided by a third party.

There are two ways to import a certificate, indicated by the **Certificate Type** option:

X.509 (PEM) Enter the **Certificate data** and **Private key data**, which are both required. See *Certificate Settings* for details on populating the contents of the fields.

The most common error is not pasting in the right portion of the certificate or private key. Make sure to include the entire block, including the beginning header and ending footer around the encoded data.

PKCS #12 (PFX) This method reads the certificate data from a PKCS #12 file, commonly found with a `.p12` extension. If the `.p12` file contains a CA, it is also imported along with the certificate, provided it does not already exist locally.

PKCS #12 Certificate Click **Browse** to locate the `.p12` file on the local client, it will be uploaded and read when saving.

PKCS #12 Certificate Password Enter the password used to protect the contents of the `.p12` file

Intermediates When set, if the PKCS #12 file contains multiple CA entries in a chain, this option will import all of them instead of only one.

Create a Certificate Signing Request

Choosing a **Method** of *Certificate Signing Request* creates a new request file that can signed by a CA at a later time, including by a third party CA not present on the firewall. This is commonly used to obtain a certificate from a trusted root certificate authority.

The parameters for creating this certificate are identical to those for creating a certificate and are covered in *Certificate Settings*.

Note: Though the GUI shows fields for **Certificate Type** and **Alternative Names** as described in *Certificate Settings*, they are only suggestions for the CA. The signing CA may ignore these options and replace them with values of its own.

Sign a Certificate Signing Request

Signing a certificate signing request (CSR) is a special process which uses an internal CA on the firewall to sign a CSR and turn it into a full-fledged certificate.

The following options are available when signing a CSR:

CA to sign with The CA on the firewall which will sign this CSR. This must be an internal CA (private key present).

CSR to sign This option chooses whether to sign a new CSR not present on the firewall or an existing CSR on the firewall.

New CSR When chosen, the GUI presents fields in which the CSR data can be pasted.

CSR Data The PEM-encoded CSR data. CSR data is typically contained in a file ending with `.req` or `.pem`. It would be plain text, and enclosed in a block such as:

```
-----BEGIN CERTIFICATE REQUEST-----
[A bunch of random-looking base64-encoded data]
-----END CERTIFICATE REQUEST-----
```

Key Data The optional PEM-encoded private key for the certificate. This is not required to sign a CSR, but may be useful, or even necessary, if the resulting certificate will be used on the firewall. For example, a private key would be required for a local service or as a user certificate used with a VPN export package.

The key data is typically in a file ending in `.key`. It would be plain text data enclosed in a block such as:

```
-----BEGIN RSA PRIVATE KEY-----
[A bunch of random-looking base64-encoded data]
-----END RSA PRIVATE KEY-----
```

Existing CSR The remaining items in the drop-down list are CSR entries which already exist on the firewall. Choose one to sign.

Certificate Lifetime The lifetime of the new certificate. See *Lifetime* for details.

Digest Algorithm The digest algorithm for the new certificate. See *Digest Algorithm* for details.

When signing a CSR, the signing CA may also give new values for **Certificate Type** and **Alternative Names** as described in *Certificate Settings*. The signing process in the GUI does not support automatically reading these values from a CSR, so set them again here.

When complete, the result is a certificate entry in the list, which can then be used or exported.

12.3.3 Edit a Certificate

To edit an existing certificate:

- Navigate to **System > Cert Manager**, **Certificates** tab
- Locate the Certificate entry in the list
- Click the  icon at the end of its row to reach the Edit page for the certificate.

The Edit page can modify some aspects of the certificate, such as:

- The **Descriptive Name** of the certificate.
- The **Certificate Data**, which may need to be replaced if the certificate was renewed by a CA off the firewall.
- The **Private key data**, which may need updated if the private key is regenerated (e.g. with a stronger key, or a different key type)

The Edit page also contains options for exporting entries with a password. See *Export Password-Protected Files* for details.

12.3.4 Export a Certificate

There are multiple methods to export certificates. The primary difference is whether or not the files will have password protection. The certificate itself does not contain private information and thus does not require protection. The private key and PKCS #12 format files do contain private information and thus can be exported in a protected manner.

Export Unprotected Files

- Navigate to **System > Cert Manager**, **Certificates** tab
- Locate the Certificate entry in the list
- Click the  icon at the end of its row to export the certificate.

  The file will download with the descriptive name of the certificate as the file name, with the extension `.crt`.
- Click the  icon to export the private key for the certificate.

  The file will download with the descriptive name of the certificate as the file name, with the extension `.key`.
- Click the  icon to export a PCKS #12 file containing the CA, certificate, and private key together.

  The file will download with the descriptive name of the certificate as the file name, with the extension `.p12`.

Export Password-Protected Files

The GUI can also export password-protected versions of the private key and PKCS #12 archives. This is more secure, but some systems may not support using password-protected keys.

- Navigate to **System > Cert Manager**, **Certificates** tab
- Locate the Certificate entry in the list
- Click the  icon at the end of its row to reach the Edit page for the certificate.
- Fill in the desired **Export Password**
- Click the  **Export Private Key** button to export the private key for the certificate.

  The password-protected file will download with the descriptive name of the certificate as the file name, with the extension `.key`.
- Click the  **PCKS #12** button to export a PCKS #12 file containing the CA, certificate, and private key together.

  The password-protected file will download with the descriptive name of the certificate as the file name, with the extension `.p12`.

12.3.5 Export a Certificate Signing Request

- Navigate to **System > Cert Manager**, **Certificates** tab
- Locate the CSR entry in the list
- Click the  icon at the end of its row to export the CSR.

  The file will download with the descriptive name of the CSR as the file name, with the extension `.req`.

12.3.6 Remove a Certificate

To remove a certificate, first it must be removed from active use.

- Check areas that can use a certificate, such as the WebGUI options, OpenVPN, IPsec, and packages

---

Note: In most cases, the areas using a certificate are noted in the **In Use** column of the certificate list. This does not necessarily include all areas, especially if the certificate is used by a package.

- Remove entries using the certificate, or choose another certificate
- Navigate to **System > Cert Manager** on the **Certificates** tab
- Locate the certificate to delete in the list
- Click  at the end of the row for the certificate
- Click OK on the confirmation dialog

12.3.7 Renew a Certificate

To renew a certificate entry:

- Navigate to **System > Cert Manager**, **Certificates** tab
- Locate the certificate entry in the list
- Click  at the end of the row for the certificate
- Follow the rest of the renewal procedure as described in *Renew or Reissue a CA or Certificate*

12.3.8 User Certificates

If a VPN is being used that requires user certificates, they may be created in one of several ways. The exact method depends on where the authentication for the VPN is being performed and whether or not the certificate already exists.

No Authentication or External Authentication

If there is no user authentication, or if the user authentication is being performed on an external server (RADIUS, LDAP, etc) then make a user certificate like any other certificate described earlier. Ensure that *User Certificate* is selected for the **Certificate Type** and set the **Common Name** to match the username.

Local Authentication

If user authentication is being performed by this firewall, the user certificate can be made inside of the User Manager. The User Manager can create a certificate while creating a user or it can add certificates to existing users. These processes are documented at *Manage Local Users*.

12.4 Renew or Reissue a CA or Certificate

When a CA or certificate expires it must be replaced, renewed, or reissued. The GUI can Renew or Reissue a certificate using a semi-automatic process. This process can retain the existing properties of the CA or certificate, but results in a freshly signed copy. This process can also make changes to the lifetime, keys, and digest so they meet current security best practices.

The new copy of this certificate must be distributed to the intended target as it was originally.

12.4.1 Certificate Properties

The Renew or Reissue page displays information about the entry, including:

Subject The subject of the certificate, containing its Distinguished Name (DN)

Serial The serial number of the certificate.

Subject Key ID Fingerprint of the the certificate key.

Certificate Type Either **User** or **Server**, if known.

Issued By The CA which signed the certificate (Name and DN)

12.4.2 Renew or Reissue Options

There are two options available which control what happens when the certificate is renewed:

Reuse Key When set (default), the existing key on the certificate is retained. When unset, a fresh key will be created when the certificate is reissued.

Strict Security When set, upgrades the security of the certificate to meet current standards.

The Renew or Reissue page performs a security analysis on the certificate, comparing its current values for Lifetime, Digest, and RSA Key size with current best security practices. This analysis is printed at the bottom of the page. If any of the values are weak, the **Would Change** column in the analysis indicates **Yes**.

12.4.3 Renew or Reissue Example

To start the renewal process, first locate the CA or certificate to renew:

- Navigate to **System > Cert Manager**
- Navigate to the **CAs** tab for CA entries, or the **Certificates** tab for certificates
- Locate the entry to renew in the list
- Click  at the end of the row for the certificate to load the Renew or Reissue page for the certificate

---

Note: The  icon only appears for entries which have been signed by an internal CA on the firewall.

- Review the contents of the page
- Set the *Renew or Reissue Options* as desired
- Click  **Renew/Reissue**
- Click **OK** to confirm the action

When the process completes, the certificate entry is updated in the configuration.

Note: If the certificate is in use by a service on the firewall, the associated service(s) are restarted automatically.

For user certificates, the updated certificate must be exported and transmitted to the user. If a new key was generated by the renewal process, it must also be transmitted to the user.

12.5 Certificate Revocation List Management

Certificate Revocation Lists (CRLs) are a part of the X.509 system that publish lists of certificates that must no longer be trusted. These certificates may have been compromised or otherwise need to be invalidated. An application using a CA, such as OpenVPN may optionally use a CRL so it can verify connecting client certificates. A CRL is generated and signed against a CA using its private key, so in order to create or add certificates to a CRL in the GUI, the private key of the CA must be present. If the CA is managed externally and the private key for the CA is not on the firewall, a CRL may still be generated outside of the firewall and imported.

The traditional way to use a CRL is to only have one CRL per CA and only add invalid certificates to that CRL. The GUI, however, supports multiple CRLs for a single CA. In OpenVPN, different CRLs may be chosen for separate VPN instances. This could be used, for example, to prevent a specific certificate from connecting to one instance while allowing it to connect to another. For IPsec, all CRLs are consulted and there is no selection as currently exists with OpenVPN.

Certificate Revocation Lists are managed from **System > Cert Manager**, on the **Certificate Revocation** tab.

From this screen CRL entries can be added, edited, exported, or deleted. The list shows all existing CRLs and an option to add a new CRL from a given CA. The screen also indicates whether the CRL is internal or external (imported), and it shows a count of how many certificates have been revoked on each CRL, and indicates if the CRL is in use.

12.5.1 Create a new Certificate Revocation List

To create a new CRL:

- Navigate to **System > Cert Manager**, on the **Certificate Revocation** tab
- Select a CA from the drop-down menu under the **Create or Import a New Certificate Revocation List**
- Click  **Add** at the end of the row to create a new CRL
- Set the **Method** to *Create an Internal Certificate Revocation List*
- Enter a **Descriptive Name** for the CRL

  This is used to identify this CRL in lists around the GUI. It's usually best to include a reference to the name of the CA and/or the purpose of the CRL.

- Enter the **Lifetime** value as a number of days for which the CRL should be valid

  The default value is `9999` days, or almost 27 and a half years.

  Note: In practice, this limit would almost never be reached as the CRL is regenerated any time the CRL is edited or when a service which utilizes a CRL is reconfigured.

- Click **Save**

The browser will be return to the CRL list, and the new entry will be shown there.

12.5.2 Import an Existing Certificate Revocation List

To import a CRL from an external source:

- Navigate to **System > Cert Manager**, on the **Certificate Revocation** tab
- Select a CA from the drop-down menu under the **Create or Import a New Certificate Revocation List**
- Click  **Add** at the end of the row to create a new CRL
- Set the **Method** to *Import an Existing Certificate Revocation List*
- Enter a **Descriptive Name** for the CRL

  This is used to identify this CRL in lists around the GUI. It's usually best to include a reference to the name of the CA and/or the purpose of the CRL.
- Enter the **CRL data**

  This is typically in a file ending in `.crl`. It would be plain text data enclosed in a block such as:

```
-----BEGIN X509 CRL-----
[A bunch of random-looking base64-encoded data]
-----END X509 CRL-----
```

- Click **Save** to finish the import process.

If an error appears, follow the on-screen instructions to correct the problem and then try again. The most common error is not pasting in the right portion of the CRL data. Make sure to enter the entire block, including the beginning header and ending footer around the encoded data.

Warning: New entries cannot be added to imported CRLs. To update an imported CRL, see *Updating an Imported Certificate Revocation List*.

12.5.3 Export a Certificate Revocation List

- Navigate to **System > Cert Manager** on the **Certificate Revocation** tab
- Locate the CRL to delete in the list
- Click the  icon

The file will download with the descriptive name of the CRL as the file name, and the extension `.crl`.

12.5.4 Delete a Certificate Revocation List

- Check areas that can use a CRL, such as OpenVPN
- Remove entries using the CRL, or choose another CRL instead
- Navigate to **System > Cert Manager** on the **Certificate Revocation** tab
- Locate the CRL to delete in the list
- Click the  icon at the end of the row for the CRL

- Click OK on the confirmation dialog

If an error appears, follow the on-screen instructions to correct the problem and then try again.

12.5.5 Revoke a Certificate

A CRL isn't useful unless it contains revoked certificates. A certificate is revoked by adding the certificate to a CRL, or by entering its serial number.

- Navigate to **System > Cert Manager** on the **Certificate Revocation** tab
- Locate the CRL to edit in the list
- Click the  icon at the end of the row for the CRL

  The GUI lists any revoked certificates on the CRL, and a control to add new ones.
- Select a **Reason** from the drop-down list to indicate why the certificate is being revoked

  This information doesn't affect the validity of the certificate it is merely informational in nature. This option may be left at the default value.
- To revoke by certificate, select the certificate(s) from the **Revoke Certificates** list

  Note: Multiple certificates can be revoked at once by selecting all of them in the list.
- To revoke by serial number, enter one or more certificate serial numbers separated by spaces in the **Revoke by Serial** field
- Click  **Add** and the certificate(s) will be added to the CRL

Note: Certificates can be revoked by selection and by serial at the same time.

After adding a certificate, the CRL will be re-written if it is currently in use by any VPN instances so that the CRL changes will be immediately active.

12.5.6 Removing a Certificate from a CRL

Certificates can be removed from the CRL when editing a CRL:

- Navigate to **System > Cert Manager** on the **Certificate Revocation** tab
- Locate the CRL to edit in the list
- Click the  icon at the end of the row for the CRL
- Find the certificate in the list and click the  icon to remove it from the CRL
- Click **OK** on the confirmation dialog

After removing a certificate, the CRL will be re-written if it is currently in use by any VPN instances so that the CRL changes will be immediately active.

12.5.7 Updating an Imported Certificate Revocation List

To update an imported CRL:

- Navigate to **System > Cert Manager** on the **Certificate Revocation** tab.
- Locate the CRL to edit in the list
- Click the  icon at the end of the row for the CRL
- Enter a new copy of the the **CRL Data**
- Click **Save**

After updating the imported CRL, it will be re-written if it is currently in use by any VPN instances so that the CRL changes will be immediately active.

12.6 DH Parameters

To put it simply, the DH parameters are extra bits of randomness that help out during the key exchange process. They do not have to match on both sides of the tunnel, and new DH parameters can be made at any time. DH parameters are not specific to a given setup in the way that certificates or keys are. There is no need to import an existing set of DH parameters because generating new parameters is a better practice.

pfSense® software ships with a default set of DH parameter files so that new firewalls do not have to spend significant CPU resources to build them when they are needed. These pre-generated parameters are stored in `/etc/dh-parameters`. Selecting a specific length in the GUI will use the DH parameter set from the corresponding file. These DH parameters are not stored in config.xml.

To generate a new set of DH parameters, which can take quite a long time depending on the hardware in use, run the following commands:

```
/usr/bin/openssl dhparam -out /etc/dh-parameters.1024 1024
/usr/bin/openssl dhparam -out /etc/dh-parameters.2048 2048
/usr/bin/openssl dhparam -out /etc/dh-parameters.4096 4096
```

CPU time used to generate the parameters increases significantly with length. For example, generating 1024-bit DH parameters only takes about 7 seconds on a C2758 CPU, but generating 2048-bit parameters takes 4 minutes, and generating 4096-bit parameters takes 10 minutes.

The GUI allows longer DH parameters to be selected if they exist in `/etc/` in the format specified above.

Supported lengths are: `1024`, `2048`, `3072`, `4096`, `7680`, `8192`, `15360`, and `16384`.

For example, to generate a new set of DH parameters of length `8192`, run:

```
/usr/bin/openssl dhparam -out /etc/dh-parameters.8192 8192
```

The **Certificate Manager** under **System > Cert Manager**, creates and maintains certificate authority (CA), certificate, and certificate revocation list (CRL) entries for use by the firewall.

Entries in the Certificate Manager are used by the firewall for purposes such as TLS for the GUI, VPNs, LDAP, various packages, and more.

12.7 Basic Introduction to X.509 Public Key Infrastructure

One authentication option for VPNs is to use X.509. An in depth discussion of X.509 and Public Key Infrastructure (PKI) is outside the scope of this documentation, and is the topic of a number of entire books for those interested in details. This chapter provides a basic understanding necessary for creating and managing certificates.

With PKI, a CA is the source of trust and is the first entity of a PKI structure. This CA then signs all of the individual certificates in a set. The certificate of the CA is used on VPN servers and clients to verify the authenticity of server and client certificates. The certificate for the CA can be used to verify signing on certificates, but not to sign certificates. Signing certificates requires the private key for the CA. The secrecy of the CA private key is what ensures the security of a PKI. Anyone with access to the CA private key can generate certificates to be used on a PKI, hence it must be kept secure. This key is never distributed to clients or servers.

Warning: Never copy more files to clients than are needed, as this may compromise the security of the PKI structure.

A certificate is considered valid if it has been trusted by a given CA. In the case of a VPN, this means that a certificate made from a specific CA would be considered valid for any VPN using that CA. For that reason the best practice is to create a unique CA for each VPN that has a different level of security. For instance, if there are two mobile access VPNs with the same security access, using the same CA for those VPNs is OK. However if one VPN is for users and another VPN is for remote management, each with different restrictions, then it is best for each VPN to have a unique CA.

Certificate revocation lists (CRLs) are lists of certificates that have been compromised or otherwise invalidated. Revoking a certificate will cause it to be considered untrusted so long as the application using the CA also uses a CRL. CRLs are generated and signed against a CA using its private key, so in order to create or add certificates to a CRL in the GUI the private key for a CA must be present.

CHAPTER

THIRTEEN

FIREWALL

One of the primary functions performed by pfSense® software is filtering traffic. This chapter covers fundamentals of firewalling, best practices, and required information necessary to configure firewall rules.

13.1 Firewall

One of the primary purposes of pfSense® software is to act as a firewall, deciding which traffic to pass or block between networks.

13.1.1 Managing Firewall Rules

Firewall rules control traffic passing through the firewall. These topics describe how to create and manage rules, plus settings related to rules.

Firewalling Fundamentals

This section deals primarily with introductory firewall concepts and lays the ground work for understanding how to configure firewall rules using pfSense® software.

Basic Terminology

Rule and *ruleset* are two terms used throughout this chapter:

Rule Refers to a single entry on the **Firewall > Rules** screen. A rule instructs the firewall how to match or handle network traffic.

Ruleset Refers to a group of rules collectively. Either *all* firewall rules as a whole, or a set of rules in a specific context such as the rules on an interface tab. The complete firewall ruleset is the sum of all user configured and automatically added rules, which are covered further throughout this chapter.

Rulesets on the **Interface** tabs are evaluated on a **first match** basis by pfSense. This means that reading the ruleset for an interface from top to bottom, the first rule that matches will be the one used by the firewall. Evaluation stops after reaching this match and then the firewall takes the action specified by that rule. Always keep this in mind when creating new rules, especially when crafting rules to restrict traffic. The most permissive rules should be toward the bottom of the list, so that restrictions or exceptions can be made above them.

Note: The **Floating** tab is the lone exception to this rule processing logic. It is covered in a later section of this chapter.

Stateful Filtering

pfSense is a stateful firewall, which means it remembers information about connections flowing through the firewall so that reply traffic can be allowed automatically. This data is retained in the **State Table**. The connection information in the state table includes the source, destination, protocol, ports, and more: Enough to uniquely identify a specific connection.

Using this mechanism, traffic need only be permitted on the interface where it **enters** the firewall. When a connection matches a pass rule the firewall creates an entry in the state table. Reply traffic to connections is automatically allowed back through the firewall by matching it against the state table rather than having to check it against rules in both directions. This includes any related traffic using a different protocol, such as ICMP control messages that may be provided in response to a TCP, UDP, or other connection.

See also:

See *Firewall Advanced* and *State Type* for more information about state options and types.

State table size

The firewall state table has a maximum size to prevent memory exhaustion. Each state takes approximately 1 KB of RAM. The default state table size in pfSense is calculated by taking about 10% of the RAM available in the firewall by default. On a firewall with 1GB of RAM, the default state table size can hold approximately 100,000 entries.

See also:

See *Large State Tables* for more information on state table sizing and RAM usage.

Each user connection typically consists of two states: One created as it enters the firewall, and one as it leaves the firewall. Therefore, with a state table size of 1,000,000, the firewall can handle approximately 500,000 user sessions actively traversing the firewall before any additional connections will be dropped. This limit can be increased as needed so long as it does not exceed the available amount of RAM in the firewall.

To increase the state table size:

- Navigate to **System > Advanced** on the **Firewall & NAT** tab
- Enter the desired number for **Firewall Maximum States**, or leave the box blank for the default calculated value. See Figure *Increased State Table Size to 2,000,000*
- Click **Save**

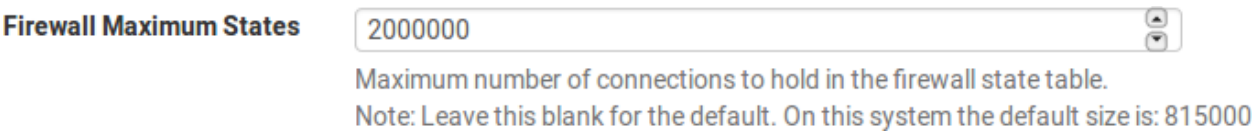


Fig. 1: Increased State Table Size to 2,000,000

Historical state table usage is tracked by the firewall. To view the graph:

- Navigate to **Status > Monitoring**
- Click  to expand the graph options
- Set **Category** for the **Left Axis** to *System*
- Set the **Graph** for the **Left Axis** to *States*

- Click  **Update Graphs**

Block vs. Reject

There are two ways to disallow traffic using firewall rules on pfSense: **Block** and **reject**.

A rule set to **block** will silently drop traffic. A blocked client will not receive any response and thus will wait until its connection attempt times out. This is the behavior of the default deny rule in pfSense.

A rule set to **reject** will respond back to the client for denied TCP and UDP traffic, letting the sender know that the connection was refused. Rejected TCP traffic receives a TCP RST (reset) in response, and rejected UDP traffic receives an ICMP unreachable message in response. Though reject is a valid choice for any firewall rule, IP protocols other than TCP and UDP are not capable of being rejected; These rules will silently drop other IP protocols because there is no standard for rejecting other protocols.

Deciding Between Block and Reject

There has been much debate amongst security professionals over the years as to the value of block vs. reject. Some argue that using block makes more sense, claiming it "slows down" attackers scanning the Internet. When a rule is set to reject, a response is sent back immediately that the port is closed, while block silently drops the traffic, causing the attacker's port scanner to wait for a response. That argument does not hold water because every good port scanner can scan hundreds or thousands of hosts simultaneously, and the scanner is not stalled waiting for a response from closed ports. There is a minimal difference in resource consumption and scanning speed, but so slight that it shouldn't be a consideration.

If the firewall blocks all traffic from the Internet, there is a notable difference between block and reject: Nobody knows the firewall is online. If even a single port is open, the value of that ability is minimal because the attacker can easily determine that the host is online and will also know what ports are open whether or not the blocked connections have been rejected by the firewall. While there isn't significant value in block over reject, we still recommend using block on WAN rules. There is some value in not actively handing information to potential attackers, and it is also a bad practice to automatically respond to an external request unnecessarily.

For rules on internal interfaces we recommend using reject in most situations. When a host tries to access a resource that is not permitted by firewall rules, the application accessing it may hang until the connection times out or the client program stops trying to access the service. With reject the connection is immediately refused and the client avoids these hangs. This is usually nothing more than an annoyance, but we still generally recommend using reject to avoid potential application problems induced by silently dropping traffic inside a network.

Introduction to the Firewall Rules screen

This section provides an introduction and overview of the Firewall Rules screen located at **Firewall > Rules**. This page lists the WAN ruleset to start with, which by default has no entries other than those for **Block private networks** and **Block bogon networks** if those options are active on the WAN interface, as shown in Figure *Default WAN Rules*.

Tip: Click  the to the right of the **Block private networks** or **Block bogon networks** rules to reach the WAN interface configuration page where these options can be enabled or disabled. (See *Block Private Networks* and *Block Bogon Networks* for more details.)

Click the **LAN** tab to view the LAN rules. By default, the only entries are the *Default allow LAN to any* rules for IPv4 and IPv6 as seen in Figure *Default LAN Rules*, and the **Anti-Lockout Rule** if it is active. The anti-lockout rule

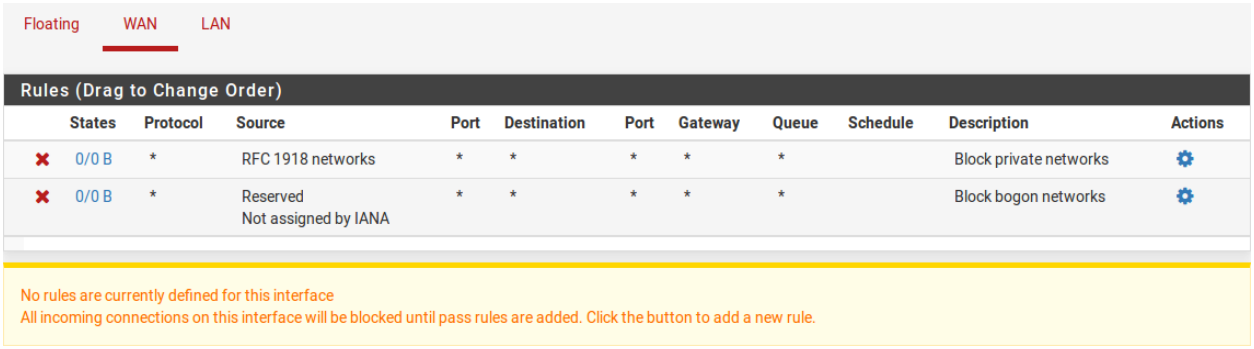


Fig. 2: Default WAN Rules

is designed to prevent administrators from accidentally locking themselves out of the GUI. Click next to the anti-lockout rule to reach the page where this rule can be disabled.

See also:

For more information on how the Anti-Lockout Rule works and how to disable the rule, see *Anti-lockout Rule* and *Anti-lockout*.

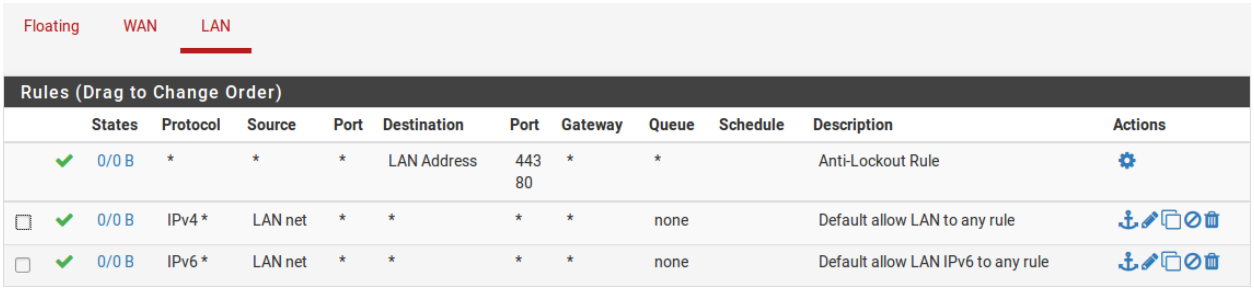


Fig. 3: Default LAN Rules

To display rules for other interfaces, click their respective tabs. OPT interfaces will appear with their descriptive names, so if the OPT1 interface was renamed `DMZ`, then the tab for its rules will also say **DMZ**.

To the left of each rule is an indicator icon showing the action of the rule: pass ( ), block ( ), or reject ( ). If logging is enabled for the rule, is shown in the same area. If the rule has any advanced options enabled, an icon is also displayed. Hovering the mouse cursor over any of these icons will display text explaining their meaning. The same icons are shown for disabled rules, except the icon and the rule are a lighter shade of their original color.

Adding a firewall rule

To add a rule to the top of the list, click  **Add**.

To add a rule to the bottom of the list, click  **Add**.

To make a new rule that is similar to an existing rule, click  to the right of the existing rule. The edit screen will appear with the existing rule's settings pre-filled, ready to be adjusted. When duplicating an existing rule, the new rule will be added directly *below* the original rule. For more information about how to configure the new rule, see *Configuring firewall rules*.

Editing Firewall Rules

To edit a firewall rule, click  to the right of the rule, or double click anywhere on the line.

The edit page for that rule will load, and from there adjustments are possible. See *Configuring firewall rules* for more information on the options available when editing a rule.

Moving Firewall Rules

Rules may be reordered in two different ways: Drag-and-drop, and using select-and-click.

To move rules using the drag-and-drop method:

- Move the mouse over the firewall rule to move, the cursor will change to indicate movement is possible.
- Click and hold the mouse button down
- Drag the mouse to the desired location for the rule
- Release the mouse button
- Click  **Save** to store the new rule order

Warning: Attempting to navigate away from the page after moving a rule, but before saving the rule, will result in the browser presenting an error confirming whether or not to exit the page. If the browser navigates away from the page without saving, the rule will still be in its original location.

To move rules in the list in groups or by selecting them first, use the select-and-click method:

- Check the box next to the left of the rules which need to be moved, or single click the rule. When the rule is selected, it will change color.
- Click  on the row **below** where the rule should be moved.

Tip: Hold **Shift** before clicking the mouse on  to move the rule below the selected rule instead of above.

When moving rules using the select-and-click method, the new order is stored automatically.

Deleting Firewall Rules

To delete a single rule, click  to the right of the rule. The firewall will present a confirmation prompt before deleting the rule.

To delete multiple rules, check the box at the start of the rows that should be removed, then click the  **Delete** button at the bottom of the list. Rules may also be selected by single clicking anywhere on their line.

Disabling and Enabling Firewall Rules

To disable a rule, click  at the end of its row. The appearance of the rule will change to a lighter shade to indicate that it is disabled and the  icon changes to  .

To enable a rule which was previously disabled, click  at the end of its row. The appearance of the rule will return to normal and the enable/disable icon will return to the original  .

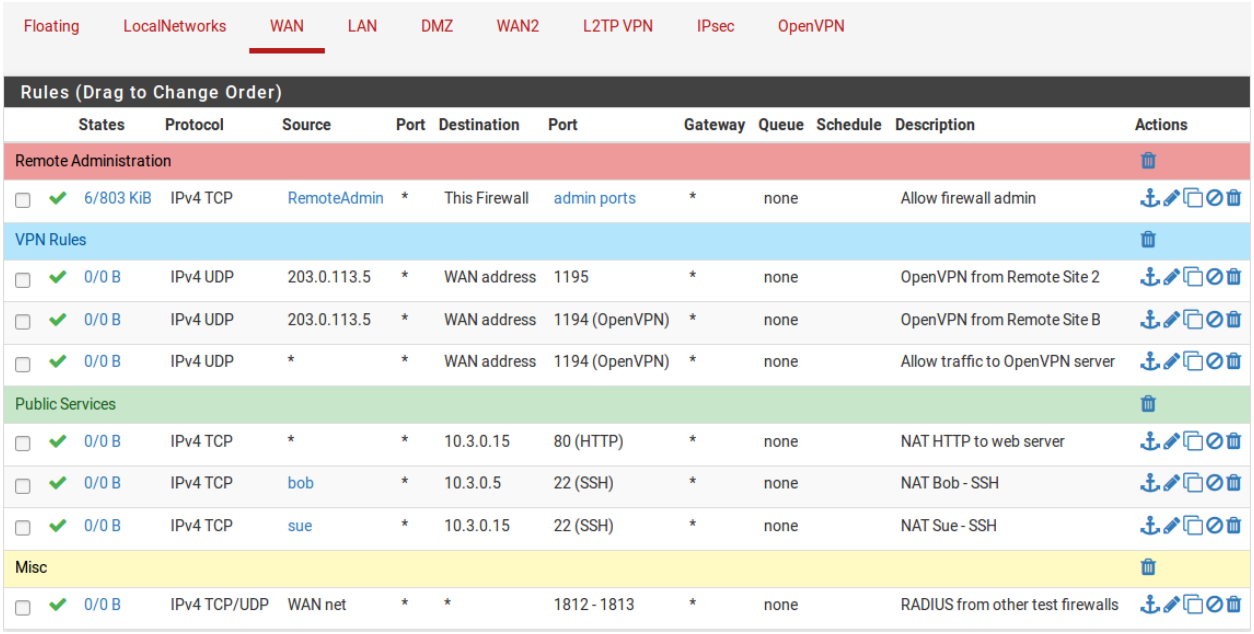
A rule may also be disabled or enabled by editing the rule and toggling the **Disabled** checkbox.

Rule Separators

Firewall Rule Separators are colored bars in the ruleset that contain a small bit of text, but do not take any action on traffic. They are useful for visually separating or adding notes to special parts of the ruleset. Figure *Firewall Rule Separators Example* shows how they can be utilize to group and document the ruleset.

To create a new Rule Separator:

- Open the firewall rule tab where the Rule Separator will reside
- Click  **Separator**
- Enter description text for the Rule Separator
- Choose the color for the Rule Separator by clicking the  icon of the desired color
- Click and drag the Rule Separator to its new location
- Click  **Save** inside the Rule Separator to store its contents



Floating | LocalNetworks | WAN | LAN | DMZ | WAN2 | L2TP VPN | IPsec | OpenVPN

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
Remote Administration											
	6/803 KiB	IPv4 TCP	RemoteAdmin	*	This Firewall	admin ports	*	none		Allow firewall admin	
VPN Rules											
	0/0 B	IPv4 UDP	203.0.113.5	*	WAN address	1195	*	none		OpenVPN from Remote Site 2	
	0/0 B	IPv4 UDP	203.0.113.5	*	WAN address	1194 (OpenVPN)	*	none		OpenVPN from Remote Site B	
	0/0 B	IPv4 UDP	*	*	WAN address	1194 (OpenVPN)	*	none		Allow traffic to OpenVPN server	
Public Services											
	0/0 B	IPv4 TCP	*	*	10.3.0.15	80 (HTTP)	*	none		NAT HTTP to web server	
	0/0 B	IPv4 TCP	bob	*	10.3.0.5	22 (SSH)	*	none		NAT Bob - SSH	
	0/0 B	IPv4 TCP	sue	*	10.3.0.15	22 (SSH)	*	none		NAT Sue - SSH	
Misc											
	0/0 B	IPv4 TCP/UDP	WAN net	*	*	1812 - 1813	*	none		RADIUS from other test firewalls	

Fig. 4: Firewall Rule Separators Example

- Click  **Save** at the bottom of the rule list

To move a Rule Separator:

- Open the firewall rule tab containing the Rule Separator
- Click and drag the Rule Separator to its new location
- Click  **Save** at the bottom of the rule list

To delete a Rule Separator:

- Open the firewall rule tab containing the Rule Separator
- Click  inside the Rule Separator on the right side
- Click  **Save** at the bottom of the rule list

Rule Separators cannot be edited. If a change in text or color is required, create a new Rule Separator and delete the existing entry.

Tracking Firewall Rule Changes

When a rule is created or updated the firewall records the user's login name, IP address, and a timestamp on the rule to track who added and/or last changed the rule in question. If the firewall automatically created the rule, that is also noted. This is done for firewall rules as well as port forwards and outbound NAT rules. An example of a rule update tracking block is shown in Figure *Firewall Rule Time Stamps*, which is visible when editing a firewall rule at the very bottom of the rule editing screen.

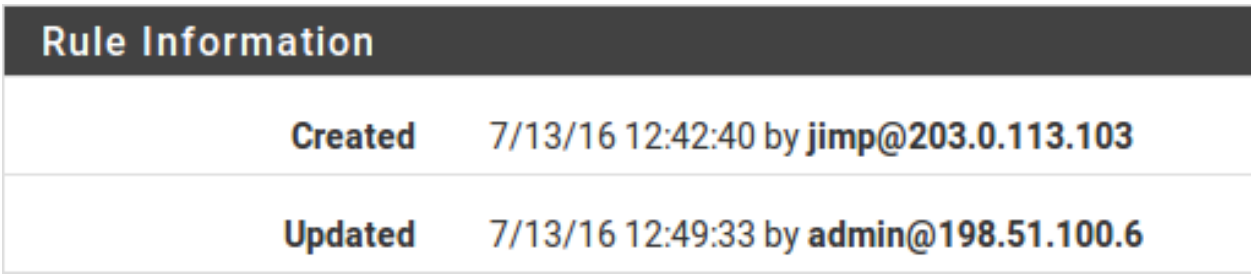


Fig. 5: Firewall Rule Time Stamps

Ingress Filtering

Ingress filtering refers to the concept of firewalling traffic entering a network from an external source such as the Internet. In deployments with multi-WAN, the firewall has multiple ingress points. The default ingress policy on pfSense® software is to **block all traffic** as there are no allow rules on WAN in the default ruleset. Replies to traffic initiated from inside the local network are automatically allowed to return through the firewall by the state table.

Egress Filtering

Egress filtering refers to the concept of firewalling traffic initiated inside the local network, destined for a remote network such as the Internet. pfSense, like nearly all similar commercial and open source solutions, comes with a LAN rule allowing everything from the LAN out to the Internet. This isn't the best way to operate, however. It has become the de facto default in most firewall solutions because it is what most people expect. The common misperception is "Anything on the internal network is 'trustworthy', so why bother filtering"?

Why employ egress filtering?

From our experience in working with countless firewalls from numerous vendors across many different organizations, most small companies and home networks do not employ egress filtering. It can increase the administrative burden as each new application or service may require opening additional ports or protocols in the firewall. In some environments it is difficult because the administrators do not completely know what is happening on the network, and they are hesitant to break things. In other environments it is impossible for reasons of workplace politics. The best practice is for administrators to configure the firewall to allow only the minimum required traffic to leave a network where possible. Tight egress filtering is important for several reasons:

Limit the Impact of a Compromised System

Egress filtering limits the impact of a compromised system. Malware commonly uses ports and protocols that are not required on most business networks. Some bots rely on IRC connections to phone home and receive instructions. Some will use more common ports such as TCP port 80 (normally HTTP) to evade egress filtering, but many do not. If access to TCP port 6667, the usual IRC port, is not permitted by the firewall, bots that rely on IRC to function may be crippled by the filtering.

Another example is a case we were involved in where the inside interface of a pfSense installation was seeing 50-60 Mbps of traffic while the WAN had less than 1 Mbps of throughput. There were no other interfaces on the firewall. Some investigation showed the cause as a compromised system on the LAN running a bot participating in a distributed denial of service (DDoS) attack against a Chinese gambling web site. The attack used UDP port 80, and in this network UDP port 80 was not permitted by the egress ruleset so all the DDoS was accomplishing was stressing the inside interface of the firewall with traffic that was being dropped. In this situation, the firewall was happily chugging along with no performance degradation and the network's administrator did not know it was happening until it was discovered by accident.

The attack described in the above paragraph likely used UDP port 80 for two main reasons:

- UDP allows large packets to be sent by the client without completing a TCP handshake. With stateful firewalls being the norm, large TCP packets will not pass until the handshake is successfully completed, and this limits the effectiveness of the DDoS.
- Those who do employ egress filtering are commonly too permissive, allowing TCP and UDP where only TCP is required, as in the case of HTTP.

These types of attacks are commonly launched from compromised web servers. With a wide open egress ruleset, the traffic will go out to the Internet, and has the potential to overflow the state table on the firewall, cost money in bandwidth usage, and/or degrade performance for everything on the Internet connection.

Outbound SMTP is another example. Only allow SMTP (TCP port 25) to leave any network from a mail server. Or if a mail server is externally hosted, only allow internal systems to talk to that specific outside system on TCP port 25. This prevents every other system in the local network from being used as a spam bot, since their SMTP traffic will be dropped. Many mail providers have moved to using only authentication submission from clients using TCP port 587, so clients should not need access to port 25. This has the obvious benefit of limiting spam, and also prevents the network from being added to numerous black lists across the Internet that will prevent that site from sending legitimate e-mail to many mail servers. This may also prevent the ISP for that site from shutting off its Internet connection due to abuse.

The ideal solution is to prevent these types of things from happening in the first place, but egress filtering provides another layer that can help limit the impact if other measures fail.

Prevent a Compromise

Egress filtering can prevent a compromise in some circumstances. Some exploits and worms require outbound access to succeed. An older but good example of this is the Code Red worm from 2001. The exploit caused affected systems to pull an executable file via TFTP (Trivial File Transfer Protocol) and then execute it. A web server almost certainly does not need to use the TFTP protocol, and blocking TFTP via egress filtering prevented infection with Code Red even on unpatched servers. This is largely only useful for stopping completely automated attacks and worms as a real human attacker will find any holes that exist in egress filtering and use them to their advantage. Again, the correct solution to prevent such a compromise is to fix the network vulnerabilities used as an attack vector, however egress filtering can help.

Limit Unauthorized Application Usage

Many applications such as VPN clients, peer-to-peer software, instant messengers, and more rely on atypical ports or protocols to function. While a growing number of peer-to-peer and instant messenger applications will port hop until finding a port which is allowed out of the local network, many will be prevented from functioning by a restrictive egress ruleset, and this is an effective means of limiting many types of VPN connectivity.

Prevent IP Spoofing

This is a commonly cited reason for employing egress filtering, but pfSense automatically blocks spoofed traffic via pf's *antispoof* functionality, so it isn't applicable here. Preventing IP Spoofing means that malicious clients cannot send traffic with obviously falsified source addresses.

Prevent Information Leaks

Certain protocols should never be allowed to leave a local network. Specific examples of such protocols vary from one environment to another, but a few common examples are:

- Microsoft RPC (Remote Procedure Call) on TCP port 135
- NetBIOS on TCP and UDP ports 137 through 139
- SMB/CIFS (Server Message Block/Common Internet File System) on TCP and UDP port 445.

Stopping these protocols can prevent information about the internal network from leaking onto the Internet, and will prevent local systems from initiating authentication attempts with Internet hosts. These protocols also fall under *Limit the Impact of a Compromised System* as discussed previously since many worms have relied upon these protocols to function. Other protocols that may be relevant are syslog, SNMP, and SNMP traps. Restricting this traffic will prevent misconfigured network devices from sending logging and other potentially sensitive information out to the Internet. Rather than worry about what protocols can leak information out of a local network and need to be blocked, the best practice is to only allow the traffic that is required.

Approaches for implementing egress filtering

On a network that has historically not employed egress filtering, it can be difficult to know what traffic is absolutely necessary. This section describes some approaches for identifying traffic and implementing egress filtering.

Allow what is known, block the rest, and work through the fallout

One approach is to add firewall rules for known required traffic to be permitted. Start with making a list of things known to be required such as in Table *Egress Traffic Required*.

Table 1: Egress Traffic Required

Description	Source	Destination	Destination port
HTTP and HTTPS from all hosts	LAN Network	Any	TCP 80 and 443
SMTP from mail server	Mail Server	Any	TCP 25
DNS queries from internal DNS servers	DNS Servers	Any	TCP and UDP 53

After making the list, configure firewall rules to pass only that traffic and let everything else hit the default deny rule.

Log Traffic and Analyze Logs

Another alternative is to enable logging on all pass rules and send the logs to a syslog server. The logs can be analyzed by the syslog server to see what traffic is leaving the network. pfSense uses a custom log format, so the logs typically need be parsed by a custom script unless the server has some knowledge of the pfSense filter log format. Analysis of the logs will help build the required ruleset with less fallout as it will yield a better idea of what traffic is necessary on the local network.

Firewall Rule Best Practices

This section covers general best practices for firewall rule configuration.

Default Deny

There are two basic philosophies in computer security related to access control: default allow and default deny. A default deny strategy for firewall rules is the best practice. Firewall administrators should configure rules to permit only the bare minimum required traffic for the needs of a network, and let the remaining traffic drop with the default deny rule built into pfSense® software. In following this methodology, the number of deny rules in a ruleset will be minimal. They still have a place for some uses, but will be minimized in most environments by following a default deny strategy.

In a default two-interface LAN and WAN configuration, pfSense utilizes default deny on the WAN and default allow on the LAN. Everything inbound from the Internet is denied, and everything out to the Internet from the LAN is permitted. All home grade routers use this methodology, as do all similar open source projects and most similar commercial offerings. It's what most people expect out of the box, therefore it is the default configuration. That said, while it is a convenient way to start, it is not the recommended means of long-term operation.

pfSense users often ask "What bad things should I block?" but that is the wrong question as it applies to a default allow methodology. Noted security professional Marcus Ranum includes default permit in his "Six Dumbest Ideas in Computer Security" paper, which is recommended reading for any security professional. Permit only what a network requires and avoid leaving the default allow all rule on the LAN and adding block rules for "bad things" above the permit rule.

Keep it short

The shorter a ruleset, the easier it is to manage. Long rulesets are difficult to work with, increase the chances of human error, tend to become overly permissive, and are significantly more difficult to audit. Utilize aliases to keep the ruleset as short as possible.

Review Firewall Rules

We recommend a manual review of the firewall rules and NAT configuration on a periodic basis to ensure they still match the minimum requirements of the current network environment. The recommended frequency of such reviews varies from one environment to another. In networks that do not change frequently, with a small number of firewall administrators and good change control procedures, quarterly or semi-annually is usually adequate. For fast changing environments or those with poor change control and several people with firewall access, review the configuration at least on a monthly basis.

Quite often when reviewing rules with customers we ask about specific rules and they respond with "We removed that server six months ago." If something else would have taken over the same internal IP address as the previous server, then traffic would have been allowed to the new server that may not have been intended.

Document The Configuration

In all but the smallest networks, it can be hard to recall what is configured where and why. We always recommend using the **Description** field in firewall and NAT rules to document the purpose of the rules. In larger or more complex deployments, create and maintain a more detailed configuration document describing the entire pfSense configuration. When reviewing the firewall configuration in the future, this will help determine which rules are necessary and why they are there. This also applies to any other area of the configuration.

It is also important to keep this document up to date. When performing periodic configuration reviews, also review this document to ensure it remains up-to-date with the current configuration. Ensure this document is updated whenever configuration changes are made.

Reducing Log Noise

By default, pfSense will log packets blocked by the default deny rule. This means all of the noise getting blocked from the Internet will be logged. Sometimes there will not be much noise in the logs, but in many environments there will inevitably be something incessantly spamming the logs.

On networks using large broadcast domains – a practice commonly employed by cable ISPs – this is most often NetBIOS broadcasts from clue-deficient individuals who connect Windows machines directly to their broadband connections. These machines will constantly pump out broadcast requests for network browsing, among other things. ISP routing protocol packets may also be visible, or router redundancy protocols such as VRRP or HSRP. In co-location environments such as data centers, a combination of all of those things may be present.

Because there is no value in knowing that the firewall blocked 14 million NetBIOS broadcasts in the past day, and that noise could be covering up logs that are important, it is a good idea to add a block rule on the WAN interface for repeated noise traffic. By adding a block rule *without logging enabled* on the WAN interface, this traffic will still be blocked, but no longer fill the logs.

The rule shown in Figure *Firewall Rule to Prevent Logging Broadcasts* is configured on a test system where the "WAN" is on an internal LAN behind an edge firewall. To get rid of the log noise to see the things of interest, we added this rule to block – but not log – anything with the destination of the broadcast address of that subnet.

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐ ✖	0/0 B	IPv4 *	*	*	10.0.64.255	*	*	none		Do not log broadcasts

Fig. 6: Firewall Rule to Prevent Logging Broadcasts

We recommend adding similar rules, matching the specifics of any log noise observed in an environment. Check the firewall logs under **Status > System Logs**, **Firewall** tab to see what kind of traffic the firewall is blocking, and review how often it appears in the log. If any particular traffic is consistently being logged more than 5 times a minute, and the traffic is not malicious or noteworthy, add a block rule for it to reduce log noise.

Logging Practices

Out of the box, pfSense does not log any passed traffic and logs all dropped traffic. This is the typical default behavior of almost every open source and commercial firewall. It is the most practical, as logging all passed traffic is rarely desirable due to the load and log levels generated. This methodology is a bit backwards, however, from a security perspective. Blocked traffic cannot harm a network so its log value is limited, while traffic that gets passed could be very important log information to have if a system is compromised. After eliminating any useless block noise as described in the previous section, the remainder is of some value for trend analysis purposes. If significantly more or less log volume than usual is observed, it is probably good to investigate the nature of the logged traffic. OSSEC, an open source host-based intrusion detection system (IDS), is one system that can gather logs from pfSense via syslog and alert based on log volume abnormalities.

Rule Methodology

In pfSense® software, rules on interface tabs are applied on a per-interface basis, always in the inbound direction on that interface. This means traffic initiated from the LAN is filtered using the LAN interface rules. Traffic initiated from the Internet is filtered with the WAN interface rules. Because all rules in pfSense are stateful by default, a state table entry is created when traffic matches an allow rule. All reply traffic is automatically permitted by this state table entry.

The exception to this is Floating rules (*Floating Rules*), which can act on any interface using the inbound, outbound, or both directions. Outbound rules are never required, because filtering is applied on the inbound direction of every interface. In some limited circumstances, such as a firewall with numerous internal interfaces, having them available can significantly reduce the number of required firewall rules. In such a case, apply egress rules for Internet traffic as outbound rules on the WAN to avoid having to duplicate them for every internal interface. The use of inbound and outbound filtering makes a configuration more complex and more prone to user error, but it can be desirable in specific applications.

Interface Groups

Interface groups, discussed in *Interface Groups*, are a method to place rules on multiple interfaces at the same time. This can simplify some rule configurations if similar rules are required on many interfaces in the same way. Interface group rules, like interface rules, are processed in the inbound direction only. The VPN tabs for OpenVPN, L2TP, and the PPPoE server are all special Interface groups that are automatically created behind the scenes.

For example, a group may be used for a collection of interfaces including all LAN or DMZ type interfaces, or for a group of VLANs.

Note: Interface groups are not effective with Multi-WAN because group rules cannot properly handle `reply-to`. Due to that deficiency, traffic matching a group rule on a WAN that does not have the default gateway will go back out the WAN with the default gateway, and not through the interface which it entered.

Rule Processing Order

So far we have talked about how the rules are processed on an interface tab, but there are three main classes of rules: Regular interface rules, Floating rules, and Interface Group rules (including VPN tab rules). The order of processing of these types is significant, and it works like so:

1. Floating Rules
2. Interface Group Rules
3. Interface Rules

The rules are ordered in that way in the actual ruleset, keep that in mind when crafting rules. For example, if an interface group contains a rule to block traffic, that rule cannot be overridden with an interface tab rule because the traffic has already been acted upon by the group rule, which was matched first in the ruleset.

The rules are processed until a match is found, however, so if a packet is *not* matched in the group rules, it can still be matched by an interface rule.

Another significant place this comes into play is with assigned OpenVPN interfaces. If an "allow all" rule is in place on the OpenVPN tab, it is matched with the group rules. This means the rules on the interface tab will not apply. This can be a problem if OpenVPN rules need to have `reply-to` in order to ensure certain traffic exits back via the VPN.

See also:

See *Ordering of NAT and Firewall Processing* for a more detailed analysis of rule processing and flow through the firewall, including how NAT rules come into play.

Automatically Added Firewall Rules

pfSense automatically adds internal firewall rules for a variety of reasons. This section describes automatically added rules and their purpose.

Anti-lockout Rule

To prevent locking an administrator out of the web interface, pfSense enables an anti-lockout rule by default. This is configurable on the **System > Advanced** page under **Anti-lockout**. This automatically added rule allows traffic from any source inside the network containing the rule, to any firewall administration protocol listening on the LAN IP address. For example, it grants access to TCP port 443 for the WebGUI, TCP port 80 for the GUI redirect, and TCP port 22 if SSH is enabled. If the WebGUI port has been changed, the configured port is the one allowed by the anti-lockout rule.

In security-conscious environments, the best practice is to disable this rule and configure the LAN rules so only an alias of trusted hosts can access the administrative interfaces of the firewall. A better practice yet is to not allow access from the LAN but only from an isolated administrative management network.

Restricting access to the administrative interface from LAN

First, to configure the firewall rules as desired to restrict access to the required management interface(s). In this typical use case example, both SSH and HTTPS are used for management, so create a ManagementPorts alias containing these ports (Figure *Alias for Management Ports*).

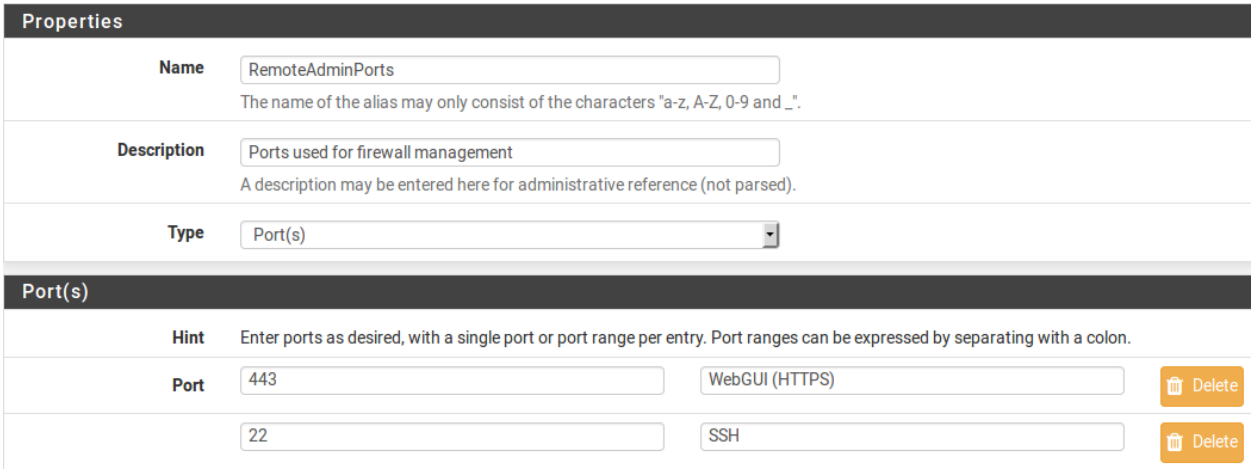


Fig. 7: Alias for Management Ports

Then create an alias for hosts and/or networks that will have access to the management interfaces (Figure *Alias For Management Hosts*).

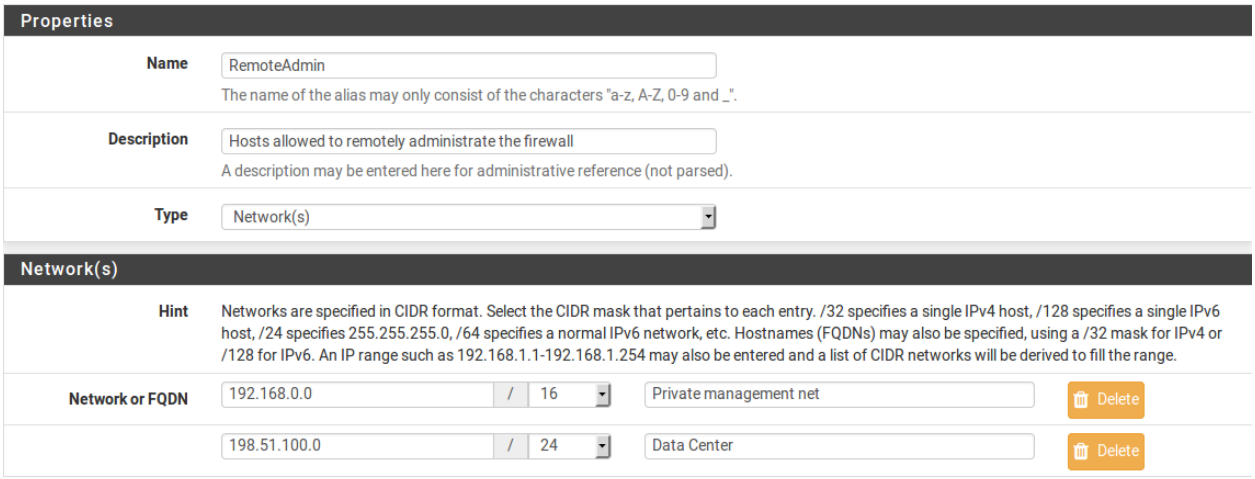


Fig. 8: Alias For Management Hosts

The resulting aliases are shown in Figure *Alias List*.

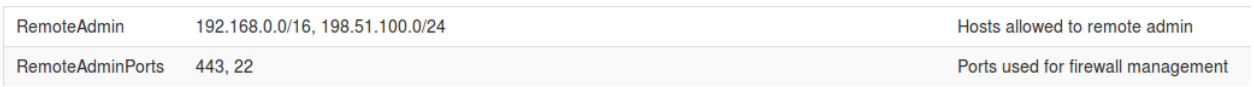


Fig. 9: Alias List

Then the LAN firewall rules must be configured to allow access by the previously defined hosts, and deny access to all else. There are numerous ways to accomplish this, depending on specifics of the environment and how egress filtering

is handled. Figure *Example Restricted Management LAN Rules* show two examples. The first allows DNS queries to the LAN IP address, which is needed if the DNS Resolver or DNS Forwarder are enabled, and also allows LAN hosts to ping the LAN IP address. It then rejects all other traffic. The second example allows access from the management hosts to the management ports, then rejects all other traffic to the management ports. Choose the methodology that works best for the network environment in question. Remember that the source port is not the same as the destination port.

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐	✔ 0/0 B	IPv4 TCP/UDP	10.0.0.0/8	*	LAN address	53 (DNS)	*	none		Allow internal network to query the DNS Resolver
☐	✔ 0/0 B	IPv4 ICMP echoreq	10.0.0.0/8	*	LAN address	*	*	none		Allow internal network to ping the LAN IP Address
☐	✔ 0/0 B	IPv4 TCP	RemoteAdmin	*	LAN address	RemoteAdminPorts	*	none		Allow access to firewall management
☐	0/0 B	IPv4 *	*	*	LAN address	*	*	none		Reject everything else to the LAN IP address
☐	✔ 0/2.59 MiB	IPv4 *	10.0.0.0/8	*	*	*	*	none		LAN Traffic

Fig. 10: Example Restricted Management LAN Rules

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐	✔ 0/0 B	IPv4 TCP	RemoteAdmin	*	LAN address	RemoteAdminPorts	*	none		Allow access to firewall management
☐	0/0 B	IPv4 TCP	*	*	LAN address	RemoteAdminPorts	*	none		Reject access to firewall management from other host
☐	✔ 0/2.59 MiB	IPv4 *	10.0.0.0/8	*	*	*	*	none		LAN Traffic

Fig. 11: Restricted Management LAN Rules Alternate Example

Once the firewall rules are configured, disable the webGUI anti-lockout rule on the **System > Advanced** page (Figure *Anti-Lockout Rule Disabled*). Check the box and click **Save**.

Note: If the management interface can no longer be accessed after disabling the anti-lockout rule, the firewall rules were not configured appropriately. Re-enable the anti-lockout rule by using the **Set Interface(s) IP address** option at the console menu, then choose to reset the LAN IP address. Set it to its current IP address, and the rule will automatically be re-enabled.

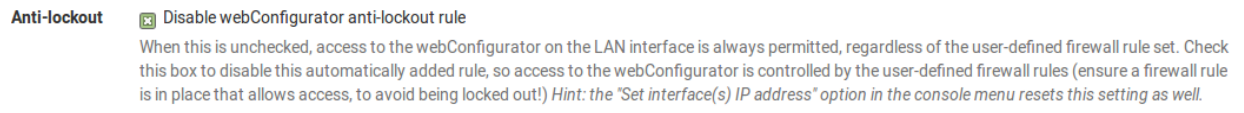


Fig. 12: Anti-Lockout Rule Disabled

Anti-spoofing Rules

pfSense uses the antispoof feature in `pf` to block spoofed traffic. This provides Unicast Reverse Path Forwarding (uRPF) functionality as defined in RFC 3704. The firewall checks each packet against its routing table, and if a connection attempt comes from a source IP address on an interface where the firewall knows that network does not reside, it is dropped. For example, a packet coming in WAN with a source IP address of an internal network is dropped. Anything initiated on the internal network with a source IP address that does not reside on the internal network is dropped.

Block Private Networks

The **Block private networks** option on the WAN interface automatically puts in a block rule for RFC 1918 subnets. Unless private IP space is in use on the WAN, enable this option. This only applies to traffic initiated on the WAN side. Local clients may still reach hosts on private networks from the inside of the firewall. This option is available for any interface, but is generally only used on WAN type interfaces. A similar rule can be created manually to block private networks on interfaces by creating an alias containing the RFC 1918 subnets and adding a firewall rule to the top of the interface rules to block traffic with a source matching that alias. (See *Private IP Addresses* for more information about private IP addresses.)

Block Bogon Networks

Bogon networks are those which should never be seen on the Internet, including reserved and unassigned IP address space. The presence of traffic from these networks can indicate either spoofed traffic or an unused subnet that has been hijacked for malicious use. Bogon lists are intended to filter invalid traffic from the Internet (e.g. on WANs) coming to the firewall for cases where the source cannot be otherwise filtered or validated, such as for public services. If rules on an interface only allow from specific remote sources, bogon blocking does not offer any benefit. pfSense software provides two bogons lists that are updated as needed, one for IPv4 bogon networks and one for IPv6 bogon networks.

Warning: Blocking bogon networks is not suited for use on local/private interfaces such as LAN. Blocking bogon networks on local interfaces can be harmful as they will block traffic which is necessary for proper local network operations, especially for IPv6. If local interfaces have proper rules which only allow from specific local sources, bogon blocking is unnecessary.

The firewall fetches an updated bogons list on the first day of each month from Netgate servers. The script runs at 3:00 a.m. local time, and sleeps a random amount of time up to 12 hours before performing the update. This list does not change frequently, and new IP address assignments are removed from the bogons list months before they are allocated, so a monthly update is adequate. To automatically update the list more frequently, change the **Update Frequency** for bogons under **System > Advanced** on the **Firewall & NAT** tab.

Note: The bogons list for IPv6 is quite large, and may not load if there is not enough memory in the system, or if the maximum number of table entries is not large enough to contain it. See *Firewall Maximum Table Entries* for information on changing that value.

See also:

For information on troubleshooting bogon updates and forcing manual updates, see *Troubleshooting Bogon Network List Updates*.

IPsec

When a site to site IPsec connection is enabled, rules are automatically added allowing the remote tunnel endpoint IP address access to UDP ports 500 and 4500, and the ESP protocol on the WAN IP address used for the connection. When IPsec for mobile clients is enabled the same traffic is allowed, but from a source of *any*, rather than a specific source address.

Because of the way policy routing works, any traffic that matches a rule specifying a gateway will be forced out to the Internet and will bypass IPsec processing. Rules are added automatically to negate policy routing for traffic destined to remote VPN subnets, but they do not always have the intended effect. To disable the automatic negation rules, see *Disable Negate rules* and add a firewall rule at the *top* of the rules on the internal interface to pass traffic to the VPN *without a gateway set*.

See also:

Automatically added IPsec rules are discussed in further depth in *IPsec*.

Default Deny Rule

Rules that do not match any user-defined rules nor any of the other automatically added rules are silently blocked by the default deny rule (as discussed in *Default Deny*).

Configuring firewall rules

When configuring firewall rules in the pfSense® WebGUI under **Firewall > Rules** many options are available to control how traffic is matched and controlled. Each of these options are listed in this section.

Action

This option specifies whether the rule will *pass*, *block*, or *reject* traffic.

Pass A packet matching this rule will be allowed to pass through the firewall. If state tracking is enabled for the rule, a state table entry is created which allows related return traffic to pass back through. See *Stateful Filtering* for more information.

Block A packet matching this rule will be discarded.

Reject A packet matching this rule will be discarded and for supported protocols, a message will be sent back to the originator indicating that the connection was refused.

See also:

See *Block vs. Reject* for a deeper description of the options and for help deciding between Block and Reject.

Disabled

To disable a rule without removing it from the rule list, check this box. It will still show in the firewall rules screen, but the rule will appear grayed out to indicate its disabled state.

Interface

The **Interface** drop down specifies the interface receiving traffic to be controlled by this rule. Remember that on interface and group tab rules, traffic is only filtered on the interface where the traffic is *initiated*. Traffic initiated from the LAN destined to the Internet or any other interface on the firewall is filtered by the *LAN* ruleset.

TCP/IP Version

Instructs the rule to apply for *IPv4*, *IPv6*, or both *IPv4+IPv6* traffic. The rules will only match and act upon packets matching the correct protocol. Aliases may be used which contain both types of IP addresses and the rule will match only the addresses from the correct protocol.

Protocol

The protocol this rule will match. Most of these options are self-explanatory. *TCP/UDP* will match both *TCP* and *UDP* traffic. Specifying *ICMP* will show an additional drop down box to select the **ICMP type**. Several other common protocols are also available.

Note: This field defaults to *TCP* for a new rule because it is a common default and it will display the expected fields for that protocol. To make the rule apply to any protocol, change this field to *any*. One of the most common mistakes in creating new rules is accidentally creating a TCP rule and then not being able to pass other non-TCP traffic such as ping, DNS, etc.

ICMP Type

When *ICMP* is selected as the protocol, this drop-down contains all possible ICMP types to match. When passing ICMP, the best practice is to only pass the required types when feasible. The most common use case is to pass only a type of *Echo Request* which will allow an ICMP ping to pass.

Tip: Historically, ICMP has a bad reputation but it is generally beneficial and does not deserve the reputation on modern networks. Allowing an ICMP type of *any* is typically acceptable when allowing ICMP.

Source

This field specifies the source IP address, subnet, or alias that will match this rule.

The drop-down box for source allows several different pre-defined types of sources:

Any Matches any address.

Single host or Alias Matches a single IP address or alias name. When this is active, an alias name may be typed in the **Source Address** field.

Network Uses both an IP address and subnet mask to match a range of addresses.

PPPoE Clients A macro that will match traffic from the client address range for the PPPoE server if the PPPoE server is enabled.

L2TP Clients A macro that will match traffic from the client address range for the L2TP server if the L2TP server is enabled.

Interface Net An entry in this list is present for each interface on the firewall. These macros specify the subnet for that interface exactly, including any IP alias VIP subnets that differ from the defined interface subnet.

Interface Address An entry in this list is present for each interface on the firewall. These macros specify the IP address configured on that interface.

Warning: The *WAN Net* choice for source or destination means the subnet of the WAN interface only. It does not mean "The Internet" or any remote host.

For rules matching TCP and/or UDP, the source port may also be specified by clicking the  **Display Advanced**. The source port is hidden behind the **Display Advanced** button because normally the source port must remain set to *any*, as TCP and UDP connections are sourced from a random port in the ephemeral port range (between 1024 through 65535, the exact range used varying depending on the OS and OS version that is initiating the connection). The source port is almost never the same as the destination port, and it should never be configured as such unless the application in use is known to employ this atypical behavior. It is also safe to define a source port as a range from `1024` to `65535`.

Selecting **Invert Match** will negate the match so that all traffic except this source value will trigger the rule.

Destination

This field specifies the destination IP address, subnet, or alias that will match this rule. See the description of the **Source** option in *Source* for more details. There is only one additional macro:

This firewall (self) Matches all IP addresses on all firewall interfaces.

For rules specifying TCP and/or UDP, the destination port, port range, or alias is also specified here. Unlike source, configuring a destination port is required in many cases, as it is more secure than using *any* and usually the destination port will be known in advance based on the protocol. Many common port values are available in the drop-down lists, or select *(other)* to enter a value manually or to use a port alias.

Tip: To specify a continuous range of ports, enter the lower port in the **From** section and the higher port value in the **To** section.

Log

This box determines whether packets that match this rule will be logged to the firewall log. Logging is discussed in more detail in *Logging Practices*.

Description

Enter a description here for reference. This is optional, and does not affect functionality of the rule. The best practice is to enter text describing the purpose of the rule. The maximum length is 52 characters.

Advanced Options

Options which are less likely to be required or that have functionality confusing to new users have been tucked away in this section of the page. Click  **Display Advanced** to show all of the advanced options. If an option in this section of the page has been set, then it will appear when the rule is loaded in the future .

Source OS

One of the more unique features of pf and thus pfSense is the ability to filter by the operating system initiating a connection. For TCP rules, pf enables passive operating system fingerprinting ("p0f") that allows rules to match based on the operating system initiating the TCP connection. The p0f feature of pf determines the OS in use by comparing characteristics of the TCP SYN packet that initiates TCP connections with a fingerprints file. Note that it is possible to change the fingerprint of an operating system to look like another OS, especially with open source operating systems such as the BSDs and Linux. This isn't easy, but if a network contains technically proficient users with administrator or root level access to systems, it is possible.

Diffserv Code Point

Differentiated Services Code Point is a way for applications to indicate inside the packets how they would prefer routers to treat their traffic as it gets forwarded along its path. The most common use of this is for quality of service or traffic shaping purposes. The lengthy name is often shortened to *Diffserv Code Point* or abbreviated as *DSCP* and sometimes referred to as the *TOS field*.

The program or device generating the packets, for example Asterisk via its `tos_sip` and `tos_audio` configuration parameters, will set the DSCP field in the packets and then it is up to the firewall and other interim routers to match and queue or act on the packets.

To match these parameters in the firewall, use the **Diffserv Code Point** drop-down entry that matches the value set by the originating device. There are numerous options, each with special meaning specific to the type of traffic. Consult the documentation for the device originating the traffic for more detail on which values must be matched.

The downside of DSCP is that it assumes routers support or act on the field, which may or may not be the case. Different routers may treat the same DSCP value in unintended or mismatched ways. Worse yet, some routers will clear the DSCP field in packets entirely as it forwards them. Also, the way pf matches traffic, the DSCP value must be set on the first packet of a connection creating a state, as each packet is not inspected individually once a state has been created.

Note: This option only reads and matches the DSCP value. It does not set a value in packets.

IP Options

Checking this box will allow packets with defined IP options to pass. By default, pf blocks all packets that have IP options set in order to deter OS fingerprinting, among other reasons. Check this box to pass IGMP or other multicast traffic containing IP options.

Disable Reply-To

The firewall adds the `reply-to` keyword to rules on WAN type interfaces by default to ensure that traffic that enters a WAN will also leave via that same WAN. In certain cases this behavior is undesirable, such as when some traffic is routed via a separate firewall/router on the WAN interface. In these cases, check this option to disable `reply-to` only for traffic matching this rule, rather than disabling `reply-to` globally.

Tag and Tagged

The **Tag** and **Tagged** fields are useful in concert with floating rules, so the firewall can mark a packet with a specific string as it enters an interface, and then act differently on a matched packet on the way out with a floating rule. See *Marking and Matching* for more on this topic.

Maximum state entries this rule can create

This option limits the maximum number of connections, total, that can be allowed by this rule. If more connections match this rule while it is at its connection limit, this rule will be skipped in the rule evaluation. If a later rule matches, the traffic has the action of that rule applied, otherwise it hits the default deny rule. Once the number of connections permitted by this rule drops below this connection limit, traffic can once again match this rule.

Maximum number of unique source hosts

This option specifies how many total source IP addresses may simultaneously connect for this rule. Each source IP address is allowed an unlimited number of connections, but the total number of distinct source IP addresses allowed is restricted to this value.

Maximum number of established connections per host

To limit access based on connections per host, use this setting. This value can limit a rule to a specific number of connections per source host (e.g. `10`), instead of a specific global connection total. This option controls how many fully established (completed handshake) connections are allowed per host that match the rule. This option is only available for use with TCP connections.

Maximum state entries per host

This setting works similar to the established count above, but it checks for state entries alone rather than tracking if a successful connection was made.

Maximum new connections / per second

This method of rate limiting helps ensure that a high TCP connection rate will not overload a server or the state table on the firewall. For example, limits can be placed on incoming connections to a mail server, reducing the burden of being overloaded by spambots. It can also be used on outbound traffic rules to set limits that would prevent any single machine from loading up the state table on the firewall or making too many rapid connections, behaviors which are common with viruses. A connection amount and a number of seconds for the time period may be configured for the rule. Any IP address exceeding the specified number of connections within the given time frame will be blocked by the firewall for one hour. Behind the scenes, this is handled by the `virusprot` table, named for its typical purpose of virus protection. This option is only available for use with TCP connections.

State timeout in seconds

Using this field, a state timeout for traffic matching this rule may be defined, overriding the default state timeout. Any inactive connections will be closed when the connection has been idle for this amount of time. The default state timeout depends on the firewall optimization algorithm in use. The optimization choices are covered in *Firewall Optimization Options*

Note: This option only controls the traffic in the inbound direction, so it is not very useful on its own. Outbound traffic for a matching connection will still have the default state timeout. To use this setting properly, a matching floating rule is also required in the outbound path taken by the traffic with a similar state timeout setting.

TCP Flags

By default, new pass rules for TCP only check for the TCP SYN flag to be set, out of a possible set of SYN and ACK. To account for more complex scenarios, such as working around asymmetric routing or other non-traditional combinations of traffic flow, use this set of controls to change how the flags are matched by the firewall rule.

The first row controls which flags must be set to match the rule. The second row defines the list of flags that will be consulted on the packet to look for a match.

The meanings of the most commonly used flags are:

SYN Synchronize sequence numbers. Indicates a new connection attempt.

ACK Indicates ACKnowledgment of data. These are replies to let the sender know data was received OK.

FIN Indicates there is no more data from the sender, closing a connection.

RST Connection reset. This flag is set when replying to a request to open a connection on a port which has no listening daemon. Can also be set by firewall software to turn away undesirable connections.

PSH Indicates that data should be pushed or flushed, including data in this packet, by passing the data up to the application.

URG Indicates that the urgent field is significant, and this packet should be sent before data that is not urgent.

To allow TCP with any flags set, check **Any Flags**.

State Type

There are three options for state tracking in pfSense that can be specified on a per-rule basis:

Keep When chosen, the firewall will create and maintain a state table entry for permitted traffic. This is the default, and the best choice in most situations.

Sloppy State Sloppy is a less strict means of keeping state that is intended for scenarios with asymmetric routing. When the firewall can only see half the traffic of a connection, the validity checks of the default state keeping will fail and traffic will be blocked. Mechanisms in pf that prevent certain kinds of attacks will not kick in during a sloppy state check.

Synproxy This option causes pfSense to proxy incoming TCP connections. TCP connections start with a three way handshake. The first packet of a TCP connection is a SYN from source, which elicits a SYN ACK response from the destination, then an ACK in return from the source to complete the handshake. Normally the host behind the firewall will handle this on its own, but synproxy state has

the firewall complete this handshake instead. This helps protect against one type of Denial of Service attack, SYN floods. This is typically only used with rules on WAN interfaces. This type of attack is best handled at the target OS level today, as every modern operating system includes capabilities of handling this on its own. Because the firewall can't know what TCP extensions the back-end host supports, when using synproxy state, it announces no supported TCP extensions. This means connections created using synproxy state will not use window scaling, SACK, nor timestamps which will lead to significantly reduced performance in most all cases. It can be useful when opening TCP ports to hosts that do not handle network abuse well, where top performance isn't a concern.

None This option will not keep state on this rule. This is only necessary in some highly specialized advanced scenarios, none of which are covered in this documentation because they are exceedingly rare.

Note: Setting *None* here only affects traffic in the inbound direction, so it is not very useful on its own since a state will still be created in the outbound direction. It must be paired with a floating rule in the outbound direction which also has the same option chosen.

No XML-RPC Sync

Checking this box prevents this rule from synchronizing to other High Availability cluster members via XMLRPC. This is covered in *High Availability*. This does not prevent a rule on a secondary node from being overwritten by the primary.

VLAN Priority (Match and Set)

802.1p, also known as IEEE P802.1p or Priority Code Point, is a way to match and tag packets with a specific quality of service priority. Unlike DSCP, 802.1p operates at layer 2 with VLANs. However, like DSCP, the upstream router must also support 802.1p for it to be useful.

There are two options in this section. The first will match an 802.1p field so the firewall can act on it. The second will inject an 802.1p tag into a packet as it passes through this firewall. Some ISPs may require an 802.1p tag to be set in certain areas, such as France, in order to properly handle voice/video/data on segregated VLANs at the correct priority to ensure quality.

There are eight levels of priority for 802.1p, and each has a two letter code in the GUI. In order from lowest priority to highest, they are:

BK Background

BE Best Effort

EE Excellent Effort

CA Critical Applications

VI Video

VO Voice

IC Internetwork Control

NC Network Control

Schedule

This option configures a schedule specifying the days and times for the rule to be in effect. Selecting "none" means the rule will always be enabled. For more information, see *Time Based Rules* later in this chapter.

Gateway

This option configures a Gateway or Gateway Group to be used by traffic matching this rule. This is covered in *Policy routing*.

In/Out Pipe (Limiters)

These selections list defined Limiters to apply a bandwidth limit to the traffic entering this interface (In) and leaving this interface (Out). More detail on limiters can be found in *Limiters*.

Ackqueue/Queue

These options define which ALTQ traffic shaper queues are applied to traffic entering and exiting this interface. For more information on traffic shaping, see *Traffic Shaper*.

Floating Rules

Floating Rules are a special type of advanced rule that can perform complicated actions not possible with rules on interface or group tabs. Floating rules can act on multiple interfaces in the inbound, outbound, or both directions. The use of inbound and outbound filtering makes designing the rules more complex and prone to user error, but they can be desirable in specific applications.

Most firewall configurations will never have floating rules, or only have them from the traffic shaper.

Precautions/Caveats

Floating rules can be a lot more powerful than other rules, but also more confusing, and it is easier to make an error that could have unintended consequences in passing or blocking traffic.

Floating rules in the inbound direction, applied to multiple WANs, will not get `reply-to` added as they would with individual interface rules, so the same problem exists here as existed with interface groups: The traffic will always exit the WAN with the default gateway, and not return properly out the WAN it entered.

Given the relative unfamiliarity of many users with Floating rules, they may not think to look there for rules when maintaining the firewall. As such, they can be a little more difficult for administration since it may not be an obvious place to look for rules.

Be careful when considering the source and destination of packets depending on the inbound and outbound direction. For example, rules in the outbound direction on a WAN would have a local source of the firewall (after NAT) and remote destination.

Potential Uses

The most common use of Floating rules is for ALTQ traffic shaping. Floating tab rules are the only type of rules which can match and queue traffic without explicitly passing the traffic.

Another way to use floating rules is to control traffic leaving from the firewall itself. Floating rules can prevent the firewall from reaching specific IP addresses, ports, and so on.

Other common uses are to ensure that no traffic can exit from other paths into a secure network, no matter what rules exist on other interfaces. By blocking outbound toward a secure network from all but the approved locations, the likelihood of later accidentally allowing traffic in through some other unintended path is reduced. Similarly, they can be used to prevent traffic destined for private networks from leaving a WAN interface, to prevent VPN traffic from leaking.

As mentioned earlier in the interface rules, they can also effectively enact state timeouts, tag/match operations, "no state" rules, and "sloppy state" rules for asymmetric routing.

Processing Order

In the inbound direction, floating rules work essentially the same as interface or group rules except that they are processed first. In the outbound direction, however, things get a little more confusing.

Firewall rules are processed after NAT rules, so rules in the outbound direction on a WAN can never match a local/private IP address source if outbound NAT is active on that interface. By the time it hits the rule, the source address of the packet is now the WAN interface IP address. In most cases this can be worked around by using the match options to tag a packet on the LAN on the way in and then matching that tag on the way out of the firewall.

Floating rules are processed before interface group rules and interface rules, so that must also be taken into consideration.

Match Action

The *match* action is unique to floating rules. A rule with the *match* action will not pass or block a packet, but only match it for purposes of assigning traffic to queues or limiters for traffic shaping. Match rules do not work with *Quick* enabled.

Quick

Quick controls whether rule processing stops when a rule is matched. The **Quick** behavior is added to all interface tab rules automatically, but on floating rules it is optional. Without *Quick* checked, the rule will only take effect if no other rules match the traffic. It reverses the behavior of "first match wins" to be "last match wins".

Using this mechanism, a default action of sorts can be crafted which will take effect only when no other rules match, similar to the default block rules on WANs.

In most situations, we advise having **Quick** selected. There are certain specific scenarios where leaving **Quick** unchecked is necessary, but they are few and far between. For most scenarios, the only rules they would have without quick selected are *match* rules traffic shaper rules.

Interface

The **Interface** selection for floating rules is different than the one for normal interface rules: It is a multi-select box so one, multiple, or all possible interfaces may be selected. Ctrl-click on interfaces to select them one by one, or use other combinations of click/drag or shift-click to select multiple interfaces.

Direction

Floating rules are not limited to the inbound direction like interface rules. They can also act in the outbound direction by selecting *out* here, or in both directions by selecting *any*. The *in* direction is also available.

The *out* direction is useful for filtering traffic from the firewall itself, for matching other undesirable traffic trying to exit an interface, or for fully configuring "sloppy state" rules, "no state" rules, or alternate state timeouts.

Marking and Matching

Using the **Tag** and **Tagged** fields, a connection can be marked by an interface tab rule and then matched in the outbound direction on a floating rule. This is a useful way to act on WAN outbound traffic from one specific internal host that could not otherwise be matched due to NAT masking the source. It can also be used similarly for applying shaping outbound on WAN from traffic specifically tagged on the way into the firewall.

For example, on a LAN rule, use a short string in the **Tag** field to mark a packet from a source of `10.3.0.56`. Then on a floating rule, quick, outbound on WAN, use **Tagged** with the same string to act on the traffic matched by the LAN rule.

Time Based Rules

Time based rules allow firewall rules to activate during specified days and/or time ranges. Time based rules function the same as any other rule, except they are effectively not present in the ruleset outside of their scheduled times.

Time Based Rules Logic

When dealing with time-based rules, the schedule determines when to apply the action specified in the firewall rule. When the current time or date is not covered by the schedule, the firewall acts as if the rule is not there. For example, a rule that passes traffic on Saturdays will only block it on other days if a separate block rule exists underneath it. The rules are processed from the top-down, the same as other firewall rules. The first match is used, and once a match is found, that action is taken if the rule is in schedule, and no other rules are evaluated.

Tip: Remember when using schedules that the rule will have **no effect** outside of their scheduled times. The rule **will not** have its action reversed because the current time is not within the scheduled time. Failing to account for this behavior could result in giving clients unintended access outside of the defined time ranges in a schedule.

Configuring Schedules for Time Based Rules

Schedules must be defined before they can be used on firewall rules. Schedules are defined under **Firewall > Schedules**, and each schedule can contain multiple time ranges. In the following example, a company wants to deny access to HTTP during business hours, and allow it all other times of the day.

Defining Times for a Schedule

To add a schedule:

- Navigate to **Firewall > Schedules**
- Click  **Add** to bring up the schedule editing screen, as seen in Figure *Adding a Time Range*.
- Enter a **Schedule Name**. This is the name that will appear in the selection list for use in firewall rules. Much like alias names, this name must only contain letters and digits, no spaces. For example: `BusinessHours`
- Enter a **Description** of this schedule, such as `Normal Business Hours`.
- Define one or more time ranges:
  - Set the **Month** by selecting a specific month and days, or by clicking the day of the week header for weekly recurring schedules.
  - Choose a **Start Time** and **Stop Time** which control when the rule is active on the selected days. The time cannot cross midnight on any day. A full day is *0:00* to *23:59*.
  - Enter an optional **Time Range Description** for this specific range, e.g. `Work Week`
  - Click  **Add Time** to add the choice as a range
  - Repeat **Month**, **Time**, and  steps for additional ranges
- Click **Save**

A schedule can apply to specific days, such as September 2, 2016, or to days of the week, such as Monday-Wednesday. To select any given day within the next year, choose the Month from the drop-down list, then click on the specific day or day numbers on the calendar. To select a day of the week, click its name in the column headers.

For this example, click on **Mon**, **Tue**, **Wed**, **Thu**, and **Fri**. This will make the schedule active for any Monday-Friday, regardless of the month. Now select the time for this schedule to be active, in 24-hour format. The hours for this example business are *9:00* to *17:00* (5pm). All times are given in the local time zone.

Once the time range has been defined, it will appear in the list at the bottom of the schedule editing screen, as in Figure *Added Time Range*.

To expand on this setup, there may be a half day on Saturday to define, or maybe the shop opens late on Mondays. In that case, define a time range for the identical days, and then another range for each day with different time ranges. This collection of time ranges will be the full schedule.

Once the schedule entry has been saved, the browser will return to the schedule list, as in Figure *Schedule List After Adding*. This schedule will now be available for use in firewall rules.

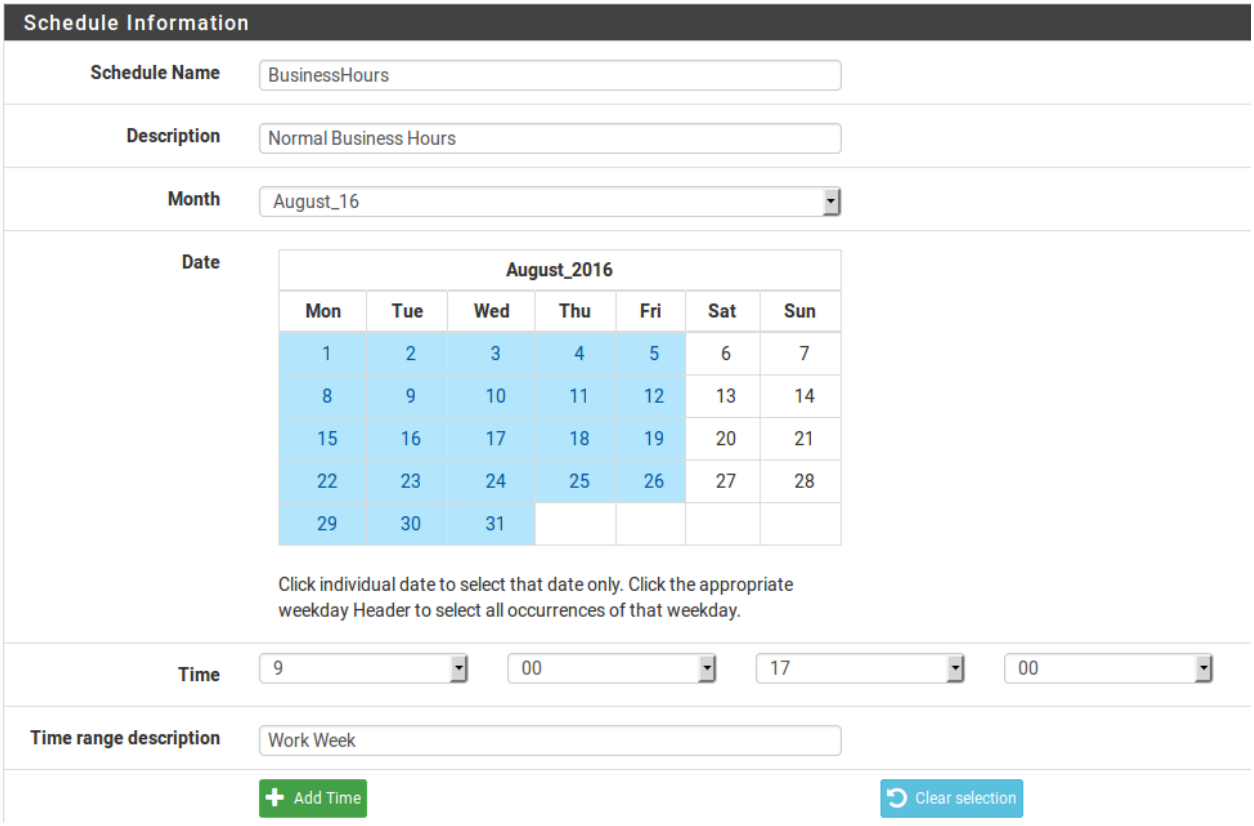


Fig. 13: Adding a Time Range

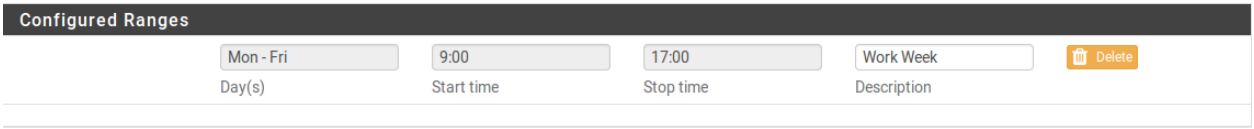


Fig. 14: Added Time Range

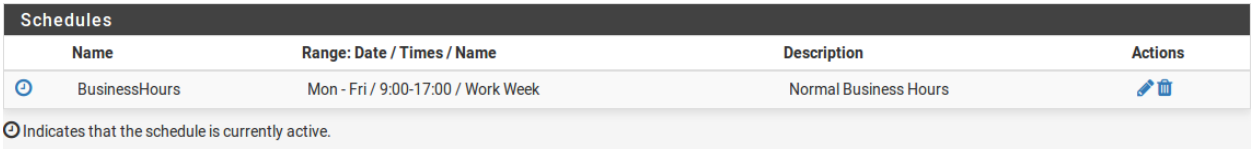


Fig. 15: Schedule List After Adding

Using the Schedule in a Firewall Rule

To create a firewall rule employing this schedule, create a new rule on the desired interface. See *Adding a firewall rule* and *Configuring firewall rules* for more information about adding and editing rules. For this example, add a rule to reject TCP traffic on the LAN interface from the LAN subnet to any destination on the HTTP port. In the advanced options for the rule, locate the **Schedule** setting and choose the *BusinessHours* schedule, as in Figure *Choosing a Schedule for a Firewall Rule*.

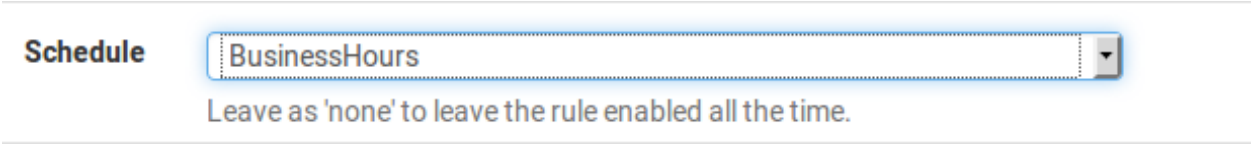


Fig. 16: Choosing a Schedule for a Firewall Rule

After saving the rule, the schedule will appear in the firewall rule list along with an indication of the schedule's active state. As shown in Figure *Firewall Rule List with Schedule*, this is a reject rule, and the schedule column indicates that the rule is currently in its active blocking state because it is being viewed at a time within the scheduled range. If the mouse cursor hovers over the schedule state indicator, a tooltip is displayed by the firewall showing how the rule will behave at the current time. Since this is being viewed inside of the times defined in the BusinessHours schedule, this will say "Traffic matching this rule is currently being denied". If there is a pass rule that would match the traffic out on port 80 from the LAN net after this rule, then it would be allowed outside of the scheduled hours.



States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0/0 B	IPv4 TCP	LAN net	*	*	80 (HTTP)	*	none	BusinessHours	Block HTTP access during business hours	

Fig. 17: Firewall Rule List with Schedule

Now that the rule is defined, test it both inside and outside of the scheduled times to ensure that the desired behavior is enacted.

Tip: By default, states are cleared for active connections permitted by a scheduled rule when the schedule expires. This shuts down access for anyone allowed by the rule while it was active. To allow these connections to remain open, check **Do not kill connections when schedule expires** under **System > Advanced** on the **Miscellaneous** tab.

Viewing the pf ruleset

pfSense® software handles translating the firewall rules in the GUI into a set of rules which can be interpreted by the packet filter (PF).

Generated Rules

The PF rules generated by the firewall configuration are in `/tmp/rules.debug`. However, that file **cannot** be edited to make persistent changes - it will be overwritten by the next filter reload event.

Note: There is rarely a need to manually edit firewall rules generated by the GUI. In most cases if it appears to be necessary, something is incorrect with the configuration.

If the generated rules truly must be edited, then the edits must be made to the source code which generates the ruleset in `/etc/inc/filter.inc`. Such changes will be lost when updating to a new version.

Interpreted Rules

PF can interpret the rules slightly differently than in the way they were generated by the filter code. To view the rule set as has been interpreted by PF, use one of the following methods.

Using the *SSH console* or *Command Prompt* field in the GUI, run the following:

Show Firewall Rules:

```
# pfctl -sr
```

Show NAT rules:

```
# pfctl -sn
```

Show all:

```
# pfctl -sa
```

For more verbose output including rule counters, ID numbers, and so on, use:

```
# pfctl -vvsr
```

There may be additional rules in anchors from packages or features such as UPnP. To view these rules, use:

```
# pfSsh.php playback pfanchordrill
```

Methods of Using Additional Public IP Addresses

Methods of deploying additional public IP addresses vary depending on how the addresses are delegated, the size of the allocation, and the goals for the specific network environment. To use additional public IP addresses with NAT, for example, the firewall will need *Virtual IP Addresses*.

There are two options for directly assigning public IP addresses to hosts: Routed public IP subnets and bridging.

Choosing between routing, bridging, and NAT

Additional public IP addresses can be put to use by directly assigning them on the systems that will use them, or by using NAT. The available options depend on how the addresses are allocated by the ISP.

Additional static IP addresses

Methods of using additional static public IP addresses vary depending on the type of assignment. Each of the common scenarios is described here.

Single IP Subnet on WAN

With a single public IP subnet on WAN, one of the public IP addresses will be on the upstream router, commonly belonging to the ISP, and another one of the IP addresses will be assigned as the WAN IP address on pfSense® software. The remaining IP addresses can be used with either NAT, bridging or a combination of the two.

To use the addresses with NAT, add Proxy ARP, IP alias or CARP type Virtual IP addresses.

To assign public IP addresses directly to hosts behind the firewall, a dedicated interface for those hosts must be bridged to WAN. When used with bridging, the hosts with the public IP addresses directly assigned must use the same default gateway as the WAN of the firewall: the upstream ISP router. This will create difficulties if the hosts with public IP addresses need to initiate connections to hosts behind other interfaces of the firewall, since the ISP gateway will not route traffic for internal subnets back to the firewall.

Figure *Multiple Public IP addresses In Use Single IP Subnet* shows an example of using multiple public IP addresses in a single block with a combination of NAT and bridging.

See also:

For information on configuration, NAT is discussed further in *Network Address Translation*, and bridging in *Bridging*.

Small WAN IP Subnet with Larger LAN IP Subnet

Some ISPs will allocate a small IP subnet as the "WAN side" assignment, sometimes called a transport or interconnect network, and route a larger "inside" subnet to the firewall. Commonly this is a /30 on the WAN side and a /29 or larger for use inside the firewall. The service provider router is assigned one end of the /30, typically the lowest IP address, and the firewall is assigned the higher IP address. The provider then routes the second subnet to the WAN IP address of the firewall. The additional IP subnet may be used by the firewall on a routed LAN or OPT interface with public IP addresses directly assigned to hosts, with NAT using Other type VIPs, or a combination of the two. Since the IP addresses are routed to the firewall, ARP is not needed so VIP entries are not necessary for use with NAT.

Because pfSense is the gateway on the local segment, routing from the public local subnet hosts to LAN is much easier than in the bridged scenario required when using a single public IP subnet. Figure *Multiple Public IP Addresses Using Two IP Subnets* shows an example that combines a routed IP subnet and NAT. Routing public IP addresses is covered in *Routing Public IP Addresses*, and NAT in *Network Address Translation*.

If the firewall is part of a High Availability cluster using CARP, the WAN side subnet will need to be a /29 so each firewall has its own WAN IP address plus a CARP VIP. The provider will route the larger inside subnet to the WAN CARP VIP in this type of configuration. The inside IP subnet must be routed to an IP address that is always available regardless of which firewall is up, and the smallest subnet usable with CARP is a /29. Such a setup with CARP is the same as illustrated above, with the OPT1 gateway being a CARP VIP, and the provider routing to a CARP VIP rather than the WAN IP address. CARP is covered in *High Availability*.

WAN IP Address Assignments	
WAN Subnet	192.0.2.128/29
WAN Gateway	192.0.2.129
Usable IP Addresses	192.0.2.129-192.0.2.134
WAN IP Address	192.0.2.130
IP Alias VIP	192.0.2.131
Bridge Host	192.0.2.134

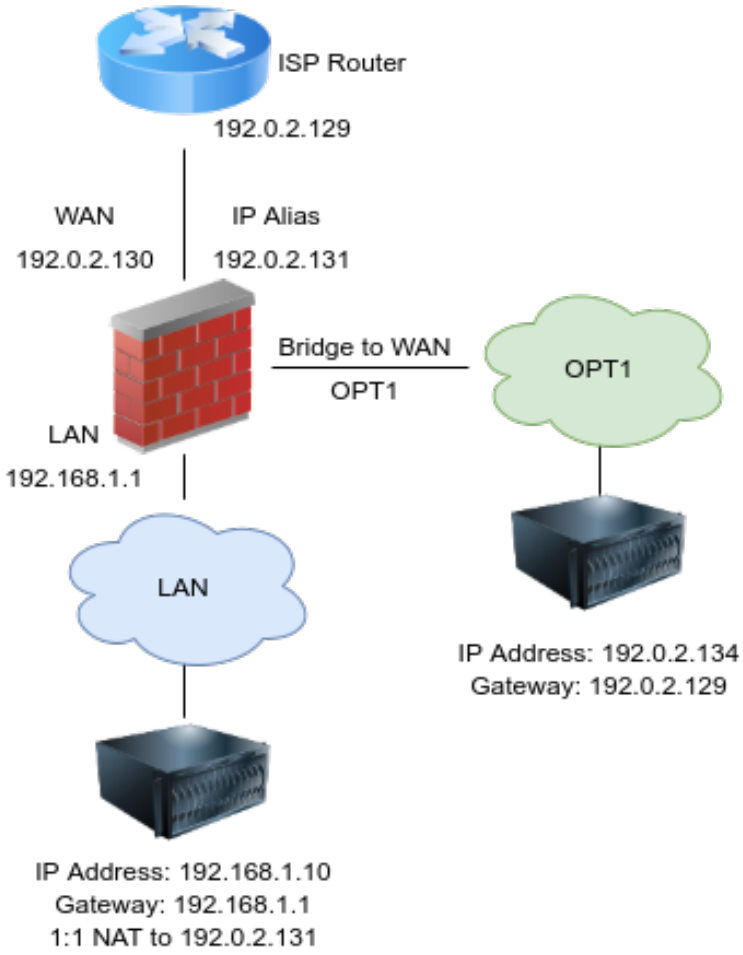


Fig. 18: Multiple Public IP addresses In Use Single IP Subnet

WAN IP Address Assignments	
WAN Subnet	192.0.2.128/30
WAN Gateway	192.0.2.129
WAN IP Address	192.0.2.130
Inside Subnet	192.0.3.0/26
OPT1 Address	192.0.3.1/26
OPT1 Usable Addresses	192.0.3.2-192.0.3.62

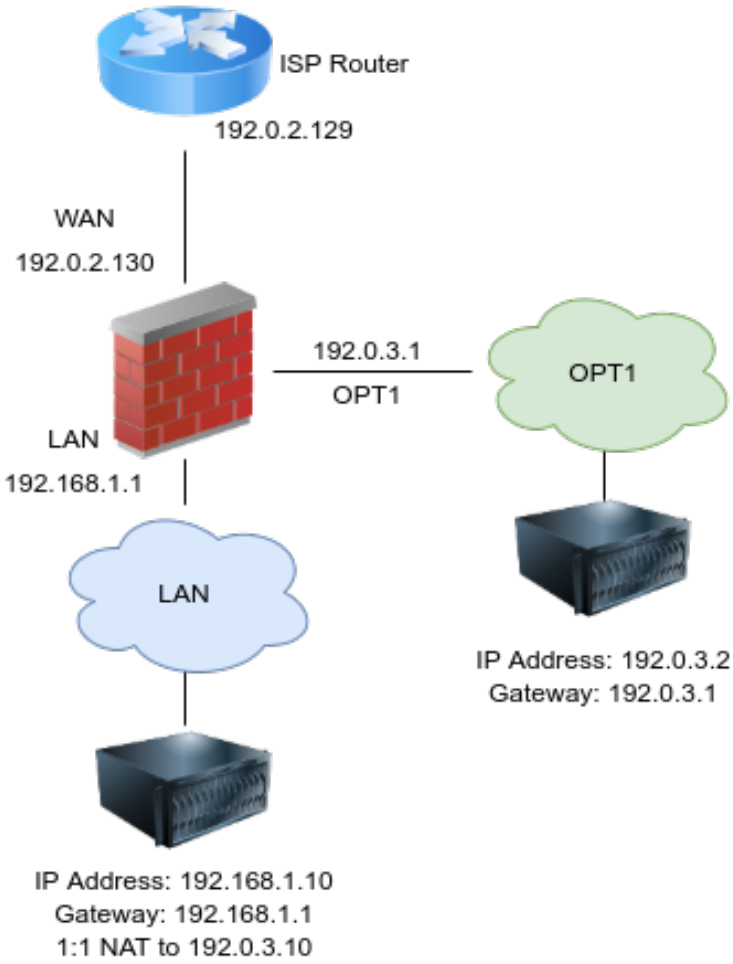


Fig. 19: Multiple Public IP Addresses Using Two IP Subnets

Multiple IP subnets

In other cases, a site may be allocated multiple IP subnets from the ISP. Usually when this happens, the site started with one of the two previously described arrangements, and later when requesting additional IP addresses the site was provided with an additional IP subnet. Ideally, this additional subnet will be routed to the firewall by the ISP, either to its WAN IP address in the case of a single firewall, or to a CARP VIP when using HA. If the provider refuses to route the IP subnet to the firewall, but rather routes it to their router and uses one of the IP addresses from the subnet as a gateway IP address, the firewall will need to use Proxy ARP VIPs, IP Alias VIPs, or a combination of IP Alias and CARP VIPs for the additional subnet. If at all possible, the provider should route the IP subnet to the firewall as it makes it easier to work with regardless of the firewall being used. It also eliminates the need to burn 3 IP addresses in the additional subnet, one for the network and broadcast addresses and one for the gateway IP address. With a routed subnet, the entire subnet is usable in combination with NAT.

Where the IP subnet is routed to the firewall, the scenario described in *Small WAN IP Subnet with Larger LAN IP Subnet* applies for an additional internal subnet. The subnet can be assigned to a new OPT interface, used it with NAT, or a combination of the two.

Additional IP Addresses via DHCP

Some ISPs require additional IP addresses to be obtained via DHCP. This is not a good means of obtaining multiple public IP addresses, and must be avoided in any serious network. A business-class connection should not require this. pfSense is one of the few firewalls which can be used in any capacity with additional IP addresses from DHCP. This offers limited flexibility in what the firewall can do with these addresses, leaving only two feasible options.

Bridging

If the additional IP addresses from DHCP must be directly assigned to the systems that will use them, bridging is the only option. Use an OPT interface bridged with WAN for these systems, and the systems must be configured to obtain their addresses using DHCP.

Pseudo multi-WAN

The only option for having the firewall pull these DHCP addresses as leases is a pseudo multi-WAN deployment. Install one network interface per public IP address, and configure each for DHCP. Plug all the interfaces into a switch between the firewall and the modem or router. Since the firewall will have multiple interfaces sharing a single broadcast domain, enable **Suppress ARP messages** on **System > Advanced**, **Networking** tab to eliminate ARP warnings in the system log, which are normal in this type of deployment.

The only use of multiple public IP addresses assigned in this fashion is for port forwarding. Port forwards can be used on each WAN interface that uses an IP address assigned to that interface by the ISP DHCP server. Outbound NAT to the OPT WANs will not work because of the limitation that each WAN must have a unique gateway IP address to properly direct traffic out of that WAN. This is discussed further in *Multiple WAN Connections*.

Virtual IP Addresses

pfSense® software enables the use of multiple IP addresses in conjunction with NAT or local services through Virtual IPs (VIPs).

There are four types of Virtual IP addresses available in pfSense: *IP Alias*, *CARP*, *Proxy ARP*, and *Other*. Each is useful in different situations. In most circumstances, pfSense will need to answer ARP request for a VIP which means that IP Alias, Proxy ARP or CARP must be used. In situations where ARP is not required, such as when additional public IP addresses are routed by a service provider to the WAN IP address on the firewall, use Other type VIPs.

pfSense will not respond to pings destined to Proxy ARP and Other type VIPs regardless of firewall rule configuration. With Proxy ARP and Other VIPs, NAT must be present on the firewall, forwarding traffic to an internal host for ping to function. See *Network Address Translation* for more information.

IP Alias

IP Aliases work like any other IP address on an interface, such as the actual interface IP address. They will respond to layer 2 (ARP) and can used as binding addresses by services on the firewall. They can also be used to handle multiple subnets on the same interface. pfSense will respond to ping on an IP Alias, and services on the firewall that bind to all interfaces will also respond on IP Alias VIPs unless the VIP is used to forward those ports in to another device (e.g. 1:1 NAT).

IP Alias VIPs can use *Localhost* as their interface to bind services using IP addresses from a block of routed addresses without specifically assigning the IP addresses to an interface. This is primarily useful in HA with CARP scenarios so that IP addresses do not need to be consumed by a CARP setup (one IP each per node, then the rest as CARP VIPs) when the subnet exists only inside the firewall (e.g. NAT or firewall services such as VPNs).

IP Aliases on their own do not synchronize to XMLRPC Configuration Synchronization peers because that would result in an IP address conflict. One exception to this is IP Alias VIPs using a CARP VIP "interface" for their interface. Those do not result in a conflict so they will synchronize. Another exception is IP Alias VIPs bound to Localhost as their interface. Because these are not active outside of the firewall itself, there is no chance of a conflict so they will also synchronize.

CARP

CARP VIPs are primarily used with High Availability redundant deployments utilizing CARP. CARP VIPs each have their own unique MAC address derived from their VHID, which can be useful even outside of a High Availability deployment.

See also:

For information on using CARP VIPs, see *High Availability*.

CARP VIPs may also be used with a single firewall. This is typically done in cases where the pfSense deployment will eventually be converted into an HA cluster node, or when having a unique MAC address is a requirement. In rare cases a provider requires each unique IP address on a WAN segment to have a distinct MAC address, which CARP VIPs provide.

CARP VIPs and IP Alias VIPs can be combined in two ways:

- To reduce the amount of CARP heartbeats by stacking IP Alias VIPs on CARP VIPs. See *Using IP Aliases to Reduce Heartbeat Traffic*.
- To use CARP VIPs in multiple subnets on a single interface. See *High Availability*.

Proxy ARP

Proxy ARP VIPs function strictly at layer 2, providing ARP replies for the specified IP address or CIDR range of IP addresses. This allows pfSense to accept traffic targeted at those addresses inside a shared subnet. For example, pfSense can forward traffic sent to an additional address inside its WAN subnet according to its NAT configuration. The address or range of addresses are not assigned to any interface on pfSense, because they don't need to be. This means no services on pfSense itself can respond on these IP addresses.

Proxy ARP VIPs do not sync to XML-RPC Configuration Sync peers because doing so would cause an IP address conflict.

Other

Other type VIPs define additional IP addresses for use when ARP replies for the IP address are not required. The only function of adding an *Other* type VIP is making that address available in the NAT configuration drop-down selectors. This is convenient when the firewall has a public IP block routed to its WAN IP address, IP Alias, or a CARP VIP.

Feature Comparison

Virtual IP Address Feature Comparison

This document summarizes and compares capabilities of the different Virtual IP Address types.

See *Virtual IP Addresses* for detailed information about each type of VIP.

VIP Features Table

Table 2: Virtual IP Address Feature Comparison

VIP Type	NAT	Binding	ARP/L2	Clustering	Subnet Mask	ICMP	Single/Range
IP Alias	Yes	Yes	Yes	See Notes	See Notes	Yes	Single
CARP	Yes	Yes	Yes	Yes	Yes	Yes	Single
Proxy ARP	Yes	No	Yes	No	n/a	No (1)	Either
Other	Yes	No	No	Yes (2)	n/a	No (1)	Either

Notes:

1. The ICMP Column represents responses from the firewall itself without NAT. With 1:1 NAT or port forwards, any VIP will pass ICMP through to the target device.
2. "Other" type VIPs are for routed subnets, and CARP is irrelevant, so they are compatible with HA (See below)

Virtual IP Feature Summary

It is difficult to express all details of VIP capabilities in a table format, so this section contains a more thorough overview of the various types and what they can/cannot do a bullet point format.

IP Alias

- **Can** be used for NAT.
- **Can** be used by the firewall itself to bind/run services.
- Adds extra IP addresses to an interface.
- Generates ARP (Layer 2) responses for the VIP address.
- Can be in a **different subnet** than the **real** interface IP address when used directly on an interface.
- **Will** respond to ICMP ping if allowed by firewall rules.
- Must be added individually
- Subnet mask should match the interface IP, or `/32`. Matching the interface subnet is best. For IP addresses in different subnets at least one IP alias VIP **must** have the correct mask for the new subnet.
- Can be stacked on top of a CARP VIP to bypass VHID limits and lower the amount of CARP heartbeat traffic.
  - Stacked IP Alias VIPs will synchronize via XMLRPC.
  - Stacked IP Alias VIPs must be inside the same subnet as the CARP VIP upon which they are placed.
- Can be added to localhost for binding services in routed subnets. IP Alias VIPs bound to localhost will synchronize via XMLRPC

CARP

- **Can** be used for NAT.
- **Can** be used by the firewall itself to bind/run services.
- Generates ARP (Layer 2) traffic for the VIP.
- Can be used for clustering (master firewall and standby failover firewall.)
- CARP VIPs may be in other subnets.
- **Will** respond to ICMP ping if allowed by firewall rules.
- Must be added individually.
- Subnet mask **must** match the interface IP address.
- Generates its own MAC address for the VIP. This MAC is different than its physical parent interface.

Proxy ARP

- **Can** be used for NAT.
- **Cannot** be used by the firewall itself to bind/run services.
- Generates ARP (Layer 2) traffic for the VIP.
- Can be in a **different subnet** than the real interface IP.
- **Will not** respond to ICMP ping.
- Can be added individually or as a subnet to make a group of VIPs.

Other

- **Can** be used for NAT.
- **Cannot** be used by the firewall itself to bind/run services.
- Can be used if the address is routed to the firewall without needing ARP/Layer 2 messages. (e.g. Upstream provider routes a subnet to the WAN IP address)
- Can be in a **different subnet** than the real interface IP address.
- **Will not** respond to ICMP echo requests.
- Can be added individually or as a subnet to make a group of VIPs.
- Can be used with CARP, e.g. subnet routed to external CARP VIP.

Using EasyRule to Add Firewall Rules

The EasyRule function found in the GUI and on the command line can add firewall rules quickly.

EasyRule in the GUI

In the pfSense® software GUI, this function is available in the Firewall Log view (**Status > System Logs**, **Firewall** tab).

The  icon next to the source IP address adds a **block** rule for that IP address on the interface. To be more precise, it creates or adds to an alias containing IP addresses added from Easy Rule and blocks them on the selected interface.

The  icon next to the destination IP address works similar to the block action, but it adds a more precise **pass** rule. This **pass** rule allows traffic on the interface but it must match the same protocol, source IP address, destination IP address, and destination port.

EasyRule in the Shell

The shell version of Easy Rule, `easyrule`, can add a firewall rule from a shell prompt. When the `easyrule` command is run without parameters, it prints a usage message to explain its syntax.

The way `easyrule` adds a block rule using an alias, or a precise pass rule specifying the protocol, source, and destination, work similar to the GUI version.

```
: easyrule
usage:
Blocking only requires an IP to block
    easyrule block <interface> <source IP>

Passing requires more detail, as it must be as specific as possible. The destination␣
↪port is
optional if the protocol does not require a port (e.g. ICMP, OSPF, etc).
    easyrule pass <interface> <protocol> <source IP> <destination ip> [destination␣
↪port]

Block example:
    easyrule block wan 1.2.3.4

Pass example (protocol with port):
    easyrule pass wan tcp 1.2.3.4 192.168.0.4 80

Pass example (protocol without port):
    easyrule pass wan icmp 1.2.3.4 192.168.0.4
```

The source code of those scripts can be adapted for adding firewall rules in other ways, but that is left as an exercise for the reader.

See also:

- *Ordering of NAT and Firewall Processing*
- *Viewing the Firewall Log*
- *Filter Reload Status*

13.1.2 Aliases

Aliases are collections of addresses that allow many hosts to be acted upon by a small number of firewall rules. They can greatly simplify a ruleset and make it easier to understand and manage.

Aliases

Aliases define a group ports, hosts, or networks. Aliases can be referenced by firewall rules, port forwards, outbound NAT rules, and other places in the firewall GUI. Using aliases results in significantly shorter, self-documenting, and more manageable rulesets.

Note: Firewall aliases are collections of entries for use by the firewall. Despite the similar names, this is different than interface IP aliases, which are a means of adding additional IP addresses to a network interface.

Aliases are located at **Firewall > Aliases**. The page is divided into separate tabs for each type of alias: **IP**, **Ports**, **URLs**, and the **All** tab which shows every alias in one large list. When creating an alias, add it to any tab and it will be sorted to the correct location based on the type chosen.

Nesting Aliases

Most aliases can be nested inside of other aliases so long as they are the same type. For example, one alias can nest an alias containing web servers, an alias containing mail servers, and a servers alias that contains both the web and mail server aliases all together in one larger `Servers` alias.

Using Hostnames in Aliases

Host and network type aliases support entries consisting of fully qualified domain name (FQDN) style hostnames (e.g. `host.domain.com`). The firewall must be able to resolve the hostname as-is using A or AAAA type DNS queries in order for these entries to function. This means that the firewall must have working DNS, and the FQDN must exist in the DNS servers used by the firewall.

Warning: This process only supports forward name resolution of FQDNs using A and AAAA records such as `host.domain.com`. Aliases **do not** support pattern matches, wildcard matches (e.g. `*.domain.com`), or any other style of record comparison.

If the DNS query for a hostname returns multiple IP addresses, all of the IP addresses returned in the result are added to the alias.

Note: This feature is *not* useful for allowing or disallowing users to large public web sites such as those served by content delivery network (CDN) providers. Such sites tend to have constantly rotating or random responses to DNS queries so the contents of the alias on the firewall do not necessarily match up with the response a user will receive when they resolve the same site name. It can work for smaller sites that have only a few servers and do not include incomplete sets of addresses in their DNS responses.

A hostname entry in a host or network type alias is periodically resolved and updated by the firewall every few minutes. The default interval is `300` seconds (5 minutes), and can be changed by adjusting the value of **Aliases Hostnames Resolve Interval** on **System > Advanced**, **Firewall & NAT** tab. This is useful for tracking dynamic DNS entries to allow specific users into services from dynamic IP addresses.

Mixing IPv4 and IPv6 Addresses in Aliases

IPv4 and IPv6 addresses can be mixed inside an alias. The firewall will use the appropriate type of addresses when the alias is referenced in a specific rule.

Alias Sizing Concerns

The total size of all tables must fit in roughly **half** the amount of **Firewall Maximum Table Entries**, which defaults to `400000`. If the maximum number of table entries is not large enough to contain all of the entries, the rules may fail to load. See *Firewall Maximum Table Entries* for information on changing that value. The aliases must fit in twice in the total area because of the way aliases are loaded and reloaded; The new list is loaded alongside the old list and then the old one is removed.

This value can be increased as much required provided that the firewall contains sufficient RAM to hold the entries. The RAM usage is similar to, but less than, the state table but it is still safe to assume approximately 1K of memory per entry to be conservative.

Alias Settings

When editing an Alias entry, the following settings are available:

Name A **Name** for the alias. The name may only consist of the characters `a-z`, `A-Z`, `0-9` and _.

Description A **Description** for the alias.

Type The **Type** for the alias, which alters the behavior of the alias and tells the firewall which types of entries can be added to the alias.

The following types are available:

Host Aliases containing single IP addresses or FQDN hostnames

Network Aliases containing CIDR-masked lists of networks, FQDN hostnames, IP address ranges, or single IP addresses

Port These aliases contain lists of port numbers or ranges of ports for TCP or UDP.

URL (IP or Port) The alias is built from the content returned by the specified URL, but is read only a single time. Once added, it becomes a normal network or port type alias.

URL Table (IP or Port) The alias is built from the content returned by the specified URL but is updated by fetching the list from the URL periodically.

Entries The lower section of the alias page contains the entries for the alias. The behavior of this section varies based on the selected alias type.

The next sections describe the behavior of each type in more detail.

Host Aliases

Host type aliases contain groups of IP addresses. For *Host* type aliases, entries are specified by IP address or fully qualified domain name (FQDN).

If an IP address range such as `192.168.1.1-192.168.1.10` or a small subnet such as `192.168.1.16/28` is entered in this field, the firewall will translate it into a list of individual IP addresses when saving the alias.

Figure *Example Hosts Alias* shows an example of a host type alias used to contain a list of public web servers.

Other host type aliases can be nested inside this entry. Hostnames may also be used as entries, as explained previously.

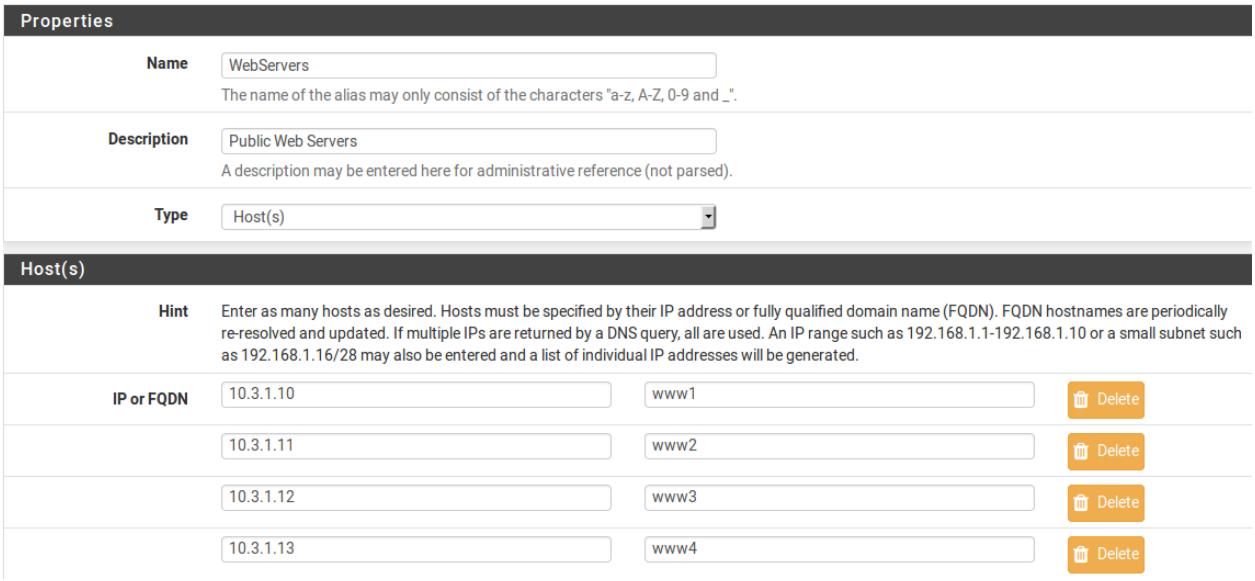


Fig. 20: Example Hosts Alias

Network Aliases

For *Network* type aliases, entries are specified in CIDR format for subnets or fully qualified domain names (FQDN) for single addresses.

For subnets, select the CIDR mask that pertains to each entry. `/32` specifies a single IPv4 host, `/128` specifies a single IPv6 host, `/24` specifies `255.255.255.0`, `/64` specifies a normal IPv6 network, etc.

Hostnames (FQDNs) may also be specified, using a `/32` mask for IPv4 or `/128` for IPv6.

Figure *Example Network Alias* shows an example of a network alias that is used later in this chapter.

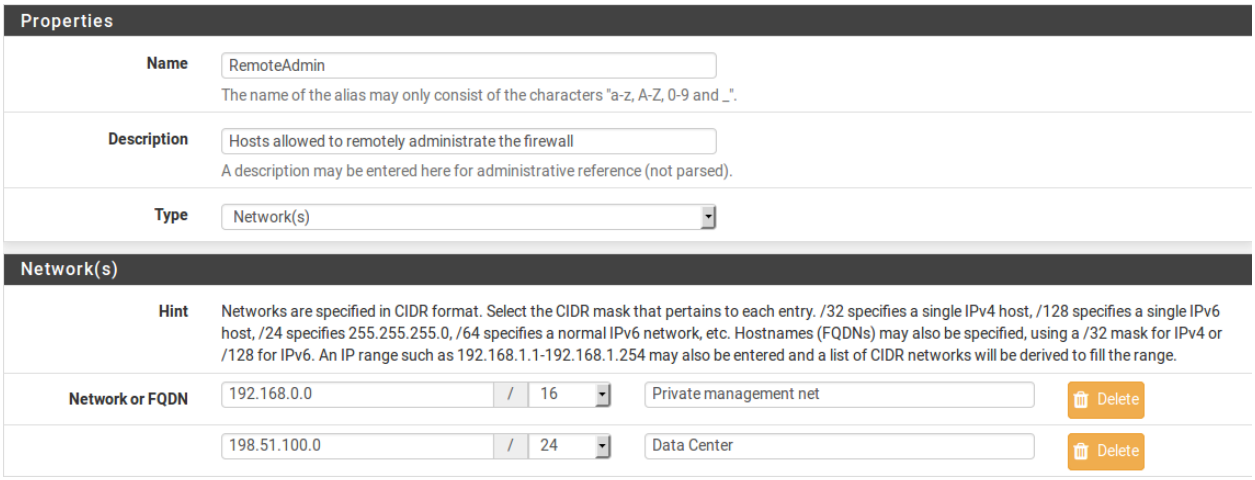


Fig. 21: Example Network Alias

Other host or network aliases can be nested inside this entry. Hostnames may also be used as entries, as explained previously.

When an alias entry contains an IPv4 range it is automatically translated by the firewall to an equivalent set of IPv4

CIDR networks that will exactly contain the provided range. As shown in Figure *Example IP Range After*, the range is expanded when the alias is saved, and the resulting list of IPv4 CIDR networks will match exactly the requested range.



Fig. 22: Example IP Range Before

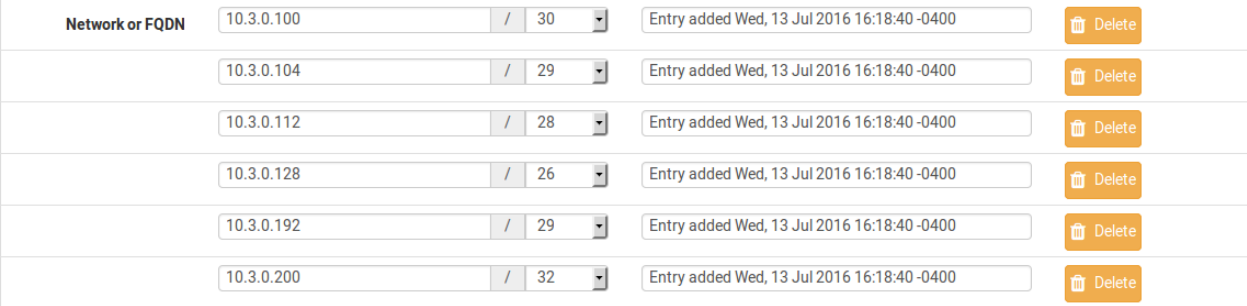


Fig. 23: Example IP Range After

Port Aliases

Port type aliases contain groups of ports and port ranges. A single port is an integer from `1-65535`. A port range is two ports separated by a colon (`:`), for example, `1194:1199` and matches the specified ports and any ports in between.

The protocol is not specified in the alias; The firewall rule where the alias is used will define the protocol as TCP, UDP, or both. Figure *Example Ports Alias* shows an example of a port type alias.

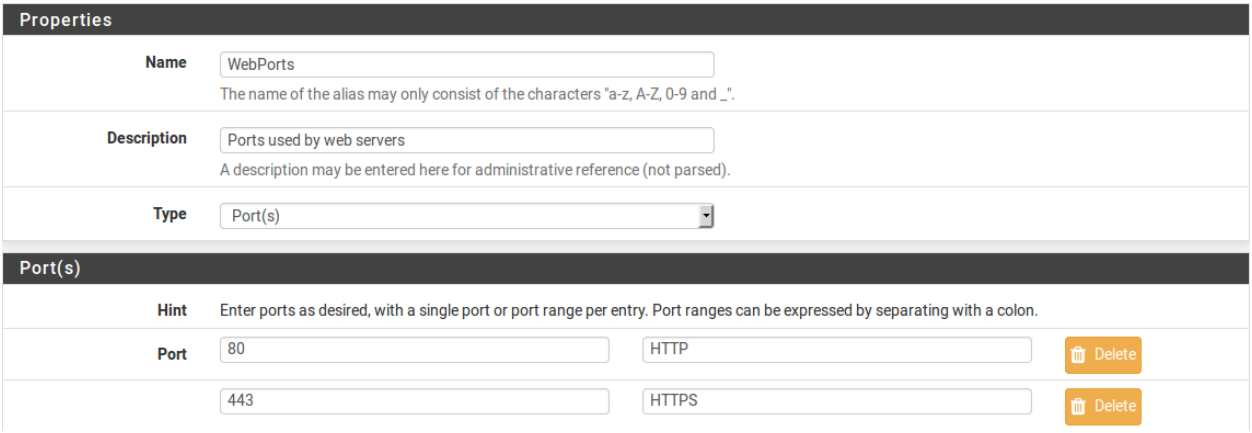


Fig. 24: Example Ports Alias

Enter another port-type alias name into the **Port** field to nest other port-type aliases inside this alias.

URL Aliases

With a URL type alias, each entry contains a URL which returns text content containing a list of entries. Multiple URLs may be entered.

When **Save** is clicked, up to 3,000 entries from each URL are read from the file and imported into a network type alias.

If *URL (IPs)* is selected, then the URLs must contain IP address or CIDR masked network entries, and the firewall creates a network type alias from the contents.

If *URL (Ports)* is selected, then the URL must contain only port numbers or ranges, and the firewall creates a port type alias from the contents.

For a URL type alias, the contents of the alias are re-fetched every 24 hours from the stored URL by the firewall.

URL Table Aliases

A URL Table alias behaves in a significantly different way than the URL alias. For starters, it does not import the contents of the file into a normal alias. It downloads the contents of the URL into a special location on the firewall and uses the contents for what is called a `persist` table, also known as a file-based alias. The full contents of the alias are not directly editable in the GUI, but can be viewed in the **Tables** viewer (See *Viewing the Contents of Tables*).

For a URL Table alias, the drop-down list after the **/** controls how many days must pass before the contents of the alias are re-fetched from the stored URL by the firewall. When the time comes, the alias contents will be updated overnight by a script which re-fetches the data.

URL Table aliases can be quite large, containing many thousands of entries. Some customers use them to hold lists of all IP blocks in a given country or region, which can easily surpass 40,000 entries. The pfBlocker package uses this type of alias when handling country lists and other similar actions.

Currently, URL Table aliases are not capable of being nested.

If *URL Table (IPs)* is selected, then the URLs must contain IP address or CIDR masked network entries, and the firewall creates a network type alias from the contents.

If *URL Table (Ports)* is selected, then the URL must contain only port numbers or ranges, and the firewall creates a port type alias from the contents.

Configuring Aliases

To add an alias:

- Navigate to **Firewall > Aliases**
- Click  **Add**
- Enter settings as described in *Alias Settings*
- Enter the type-specific information as needed. Each type has an data field and a description field for each entry.

To add new members to an alias, click  **Add** at the bottom of the list of entries.

To remove members from an alias, click  **Delete** at the end of the row to remove.

When the alias is complete, click **Save** to store the alias contents.

Each manually entered alias is limited to 5,000 members, but some browsers have trouble displaying or using the page with more than around 3,000 entries. For large numbers of entries, use a *URL Table* type alias which is capable of handling larger lists.

Bulk Import Network Aliases

Another method of importing multiple entries into an alias is to use the bulk import feature.

To use the import feature:

- Navigate to **Firewall > Aliases**
- Click  **Import**
- Fill in the **Alias Name** and **Description**
- Enter the alias contents into the **Aliases to import** text area, one entry per line.
- Click **Save**

Common usage examples for this page include lists of IP addresses, networks, and blacklists. The list may contain IP addresses, CIDR masked networks, IP ranges, or port numbers. The firewall will attempt to determine the target alias type automatically.

The firewall imports items into a normal alias which can be edited later.

Using Aliases

When a letter is typed into an input box which supports aliases, the GUI displays a list of matching aliases. Select the desired alias from the list, or type its name out completely.

Note: Alias autocompletion is not case sensitive but it is restricted by type. For example, a Network or Host type alias will be listed in autocomplete for a Network field, but a Port alias will not; A port alias can be used in a port field, but a Network alias will not be in the list.

Figure *Autocompletion of Hosts Alias* shows how the `WebServers` alias, configured as shown in Figure *Example Hosts Alias*, can be used in the **Destination** field when adding or editing a firewall rule.

- Edit the firewall rule
- Select *Single host or alias*
- Then type the first letter of the desired alias: Enter `W` and the alias appears as shown.



Fig. 25: Autocompletion of Hosts Alias

Figure *Autocompletion of Ports Alias* shows the autocompletion of the ports alias configured as shown in Figure *Example Ports Alias*. If multiple aliases match the letter entered, all matching aliases of the appropriate type are listed. Click on the desired alias to select it.

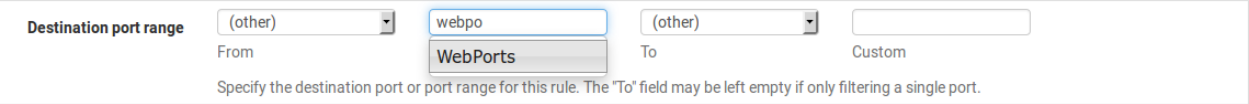


Fig. 26: Autocompletion of Ports Alias

Figure *Example Rule Using Aliases* shows the rule created using the `WebServers` and `WebPorts` aliases. This rule is on WAN, and allows any source to the IP addresses defined in the `WebServers` alias when using the ports defined in the `WebPorts` alias.



Fig. 27: Example Rule Using Aliases

Hovering the mouse cursor over an alias on the **Firewall > Rules** page shows a tooltip displaying the contents of the alias with the descriptions included in the alias. Figure *Hovering Shows Hosts Contents* shows this for the `WebServers` alias and Figure *Hovering Shows Ports Contents* for the ports alias.

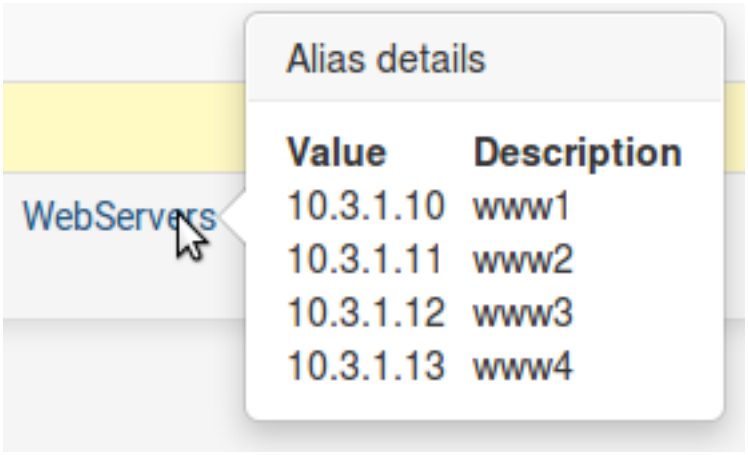


Fig. 28: Hovering Shows Hosts Contents

13.1.3 Firewall Guides

How to perform various tasks with firewall rules.

See also:

- *Blocking Web Sites*
- *Allowing Remote Access to the GUI*
- *Preventing RFC1918 Traffic from Exiting a WAN Interface*
- *Configuring pfSense Software for Online Gaming*

Troubleshooting problems with firewall behavior.

See also:

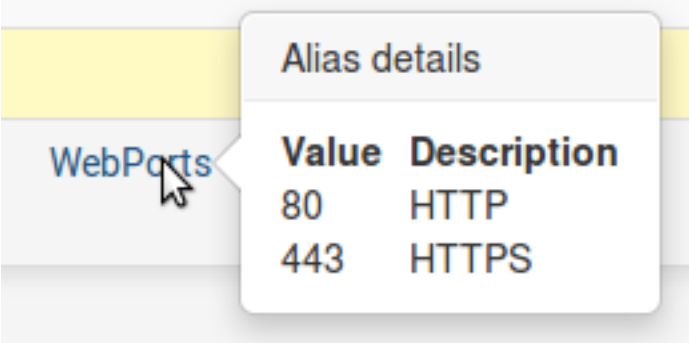


Fig. 29: Hovering Shows Ports Contents

- *Troubleshooting Thread Errors with Hostnames in Aliases*
- *Troubleshooting Firewall Rules*
- *Troubleshooting Asymmetric Routing*
- *Troubleshooting Blocked Log Entries for Legitimate Connection Packets*

CHAPTER

FOURTEEN

NETWORK ADDRESS TRANSLATION

14.1 Port Forwards

Port forwards allow access to a specific port, port range or protocol on a privately addressed internal network device. The name "port forward" was chosen because it is what most people understand in this context, and it was renamed from the more technically appropriate "Inbound NAT" to be more user-friendly. Similar functionality is also called "Destination NAT" in other products. However, "Port Forward" a misnomer, as port forward rules can redirect entire protocols such as GRE or ESP in addition to TCP and UDP ports, and it can be used for various types of traffic redirection as well as traditional port forwards. This is most commonly used when hosting servers, or using applications that require inbound connections from the Internet.

See also:

pfSense Hangouts on Youtube to view the May 2016 hangout for NAT on pfSense 2.3, The June 2016 hangout on Connectivity Troubleshooting, and the December 2013 Hangout on Port Forward Troubleshooting, among others.

14.1.1 Risks of Port Forwarding

In a default configuration, pfSense® software does not allow any traffic initiated from hosts on the Internet. This provides protection from anyone scanning the Internet looking for systems to attack. When a port forward rule exists, pfSense will allow any traffic matching corresponding firewall rules. The firewall does not know the difference between a packet with a malicious payload and one that is benign. If the connection matches the firewall rule, it is allowed. Host based controls must be used by the target system to secure any services allowed through the firewall.

14.1.2 Port Forwarding and Local Services

Port forwards take precedence over services running locally on the firewall, such as the web interface, and SSH. For example this means if remote web interface access is allowed from the WAN using HTTPS on TCP port 443, a port forward on WAN for TCP 443 will take precedence and the web interface will no longer be accessible from WAN. This does not affect access on other interfaces, only the interface containing the port forward.

14.1.3 Port Forwarding and 1:1 NAT

Port forwards also take precedence over 1:1 NAT. If a port forward is defined on one external IP address forwarding a port to a host, and a 1:1 NAT entry is also defined on the same external IP address forwarding everything into a different host, then the port forward remains active and continues forwarding to the original host.

14.1.4 Port Forward Settings

When creating or editing a port forward entry, the following settings are available:

Disable A checkbox to optionally Disable this NAT port forward. To deactivate the rule, check this box.

No RDR (NOT) Negates the meaning of this port forward, indicating that **no** redirection should be performed if this rule is matched. Most configurations will not use this field. This would be used to override a forwarding action, which may be needed in some cases to allow access to a service on the firewall on an IP address being used for 1:1 NAT, or another similar advanced scenario.

Interface The interface where the port forward will be active. In most cases this will be WAN. For additional WAN links or local redirects this may be different interface. The **Interface** is the location on the firewall where traffic for this port forward enters.

Address Family The address family for the IP address on which this port will be forwarded, either *IPv4* or *IPv6*.

When an interface contains addresses of both families, the appropriate address will be used. Additionally, when selecting an interface it must have address which matches this type. When selecting a specific IP address, the address family must match the selected address.

Protocol The **Protocol** of the incoming traffic to match. This must be set to match the type of service being forwarded, whether it is *TCP*, *UDP*, or another available choice.

Most common services are *TCP* or *UDP*, but consult the documentation for the service or even a quick web search to confirm the answer. The *TCP/UDP* option forwards both TCP and UDP together in a single rule.

Source These options are hidden behind an **Advanced** button by default, and set to *any* source. The **Source** options restrict which source IP addresses and ports can access this port forward entry. These are not typically necessary.

If the port forward must be reachable from any location on the Internet, the source must be *any*. For restricted access services, use an alias here so only a limited set of IP addresses may access the port forward.

Unless the service absolutely requires a specific source port, the Source Port Range must be left as *any* since nearly all clients will use randomized source ports.

Destination The IP address where the traffic to be forwarded is initially destined. For port forwards on WAN, in most cases this is *WAN Address*. Where multiple public IP addresses are available, it may be a Virtual IP (see *Virtual IP Addresses*) on WAN.

If **Invert Match** is checked, the port forward will match any packet which does not match the specified destination instead.

Destination port range The original destination port of the traffic, as it is coming in from the Internet, before it is redirected to the specified target host.

Note: If forwarding a single port, enter it in the **From port** box and leave the **To port** box blank.

A list of common services is available to choose from in the drop down boxes in this group. Port aliases may also be used here to forward a set of services. If an alias is used here, the same alias must be used as the **Redirect target port**.

Redirect target IP The IP address where traffic will be forwarded, or technically redirected. When using an IPv6 target, it must be of the same scope as the destination.

Note: When using an alias as a value for this field, it should only contain a **single IP address**. Using multiple addresses will result in traffic being redirected to the target hosts in a round-robin fashion, but it is not ideally suited to that task. If one of the target hosts is down, traffic will still be forwarded to the unreachable target.

For situations requiring forwarding to multiple hosts, such as load balancing or failover scenarios, use the *HAProxy package*.

Redirect target port Where the forwarded port range will begin. If a range of ports is forwarded, e.g. `19000-19100`, only the local starting point is specified since the number of ports must match up one-to-one.

This field allows opening a different port on the outside than the host on the inside is listening on. For example external port `8888` may forward to local port `80` for HTTP on an internal server. A list of common services is available to pick from in the drop down box.

Port aliases may also be used here to forward a set of services. If an alias is used here, the same alias must be used as the **Destination port range**.

Description As in other parts of pfSense, this field is available for a short sentence about what the port forward does or why it exists.

No XML-RPC Sync This option is only relevant if an HA Cluster configuration is in use, and should be skipped otherwise. When using an HA cluster with configuration synchronization, checking this box will prevent the rule from being synchronized to the other members of a cluster (see *High Availability*). Typically all rules should synchronize, however. This option is only effective on master nodes, it does *not* prevent a rule from being overwritten on slave nodes.

NAT Reflection This topic is covered in more detail later in this chapter (*NAT Reflection*). This option allows reflection to be enabled or disabled a per-rule basis to override the global default. The options in this field are explained in more detail in *NAT Reflection*.

Filter Rule Association This final option is *very* important. A port forward entry only defines which traffic will be redirected, a firewall rule is required to *pass* any traffic through that redirection. By default, *Add associated filter rule* is selected. The available choices are:

None If this is chosen, no firewall rule will be created.

Add associated filter rule This option creates a firewall rule that is linked to this NAT port forward rule. Changes made to the NAT rule are updated in the firewall rule automatically. If this option is chosen, after the rule is saved a link is placed here which leads to the associated firewall rule.

This is the default behavior and the best choice for most use cases.

Add unassociated filter rule This option creates a firewall rule that separate from this NAT port forward. Changes made to the NAT rule must be manually changed in the firewall rule. This can be useful if other options or restrictions must be set on the firewall rule rather than the NAT rule.

Pass This choice uses a special `pf` keyword on the NAT port forward rule that causes traffic to be passed through without the need of a firewall rule. Because no separate

firewall rule exists, any traffic matching this rule is forwarded in to the target system.

Note: Rules using *Pass* will only work on the interface containing the default gateway for the firewall, they do not work with Multi-WAN.

14.1.5 Adding Port Forwards

Port Forwards are managed at **Firewall > NAT**, on the **Port Forward** tab. The rules on this screen are managed in the same manner as firewall rules (see *Introduction to the Firewall Rules screen*).

To add a port forward entry:

- Navigate to
- Click  **Add** button to reach the **Port Forward** editing screen
- Enter the options for the port forward as described in *Port Forward Settings*
- Click **Save**
- Click **Apply Changes**

Figure *Port Forward Example* contains an example of the port forward editing screen filled in with the proper settings to forward HTTP (port `80`) inbound on WAN destined to the WAN IP address to the internal system at `10.3.0.15`.

After clicking **Save**, the port forward list is displayed again, and the newly created entry will be present in the list, as in Figure *Port Forward List*.

Double check the firewall rule, as seen under **Firewall > Rules** on the tab for the interface upon which the port forward was created. The rule will show that traffic is allowed into the internal IP address on the proper port, as shown in Figure *Port Forward Firewall Rule*.

The *Source* of the automatically generated rule should be restricted where possible. For things such as mail and web servers that typically need to be widely accessible, this isn't practical, but for remote management services such as SSH, RDP and others, there are likely only a small number of hosts that should be able to connect using those protocols into a server from across the Internet. A much more secure practice is to create an alias of authorized hosts, and then change the source from *any* to the alias. Otherwise, the server is wide open to the entire Internet. Test the port forward first with the unrestricted source, and after verifying it works, restrict the source as desired.

If everything looks right, the port forward will work when tested from outside the network. If something went wrong, see *Port Forward Troubleshooting* later in this chapter.

14.1.6 Tracking Changes to Port Forwards

As mentioned in Figure *Firewall Rule Time Stamps* for firewall rules, a timestamp is added to a port forward entry when it is created or last edited, to show which user created the rule, and the last person to edit the rule. Firewall rules automatically created by associated NAT rules are also marked as such on the associated firewall rule's creation timestamp.

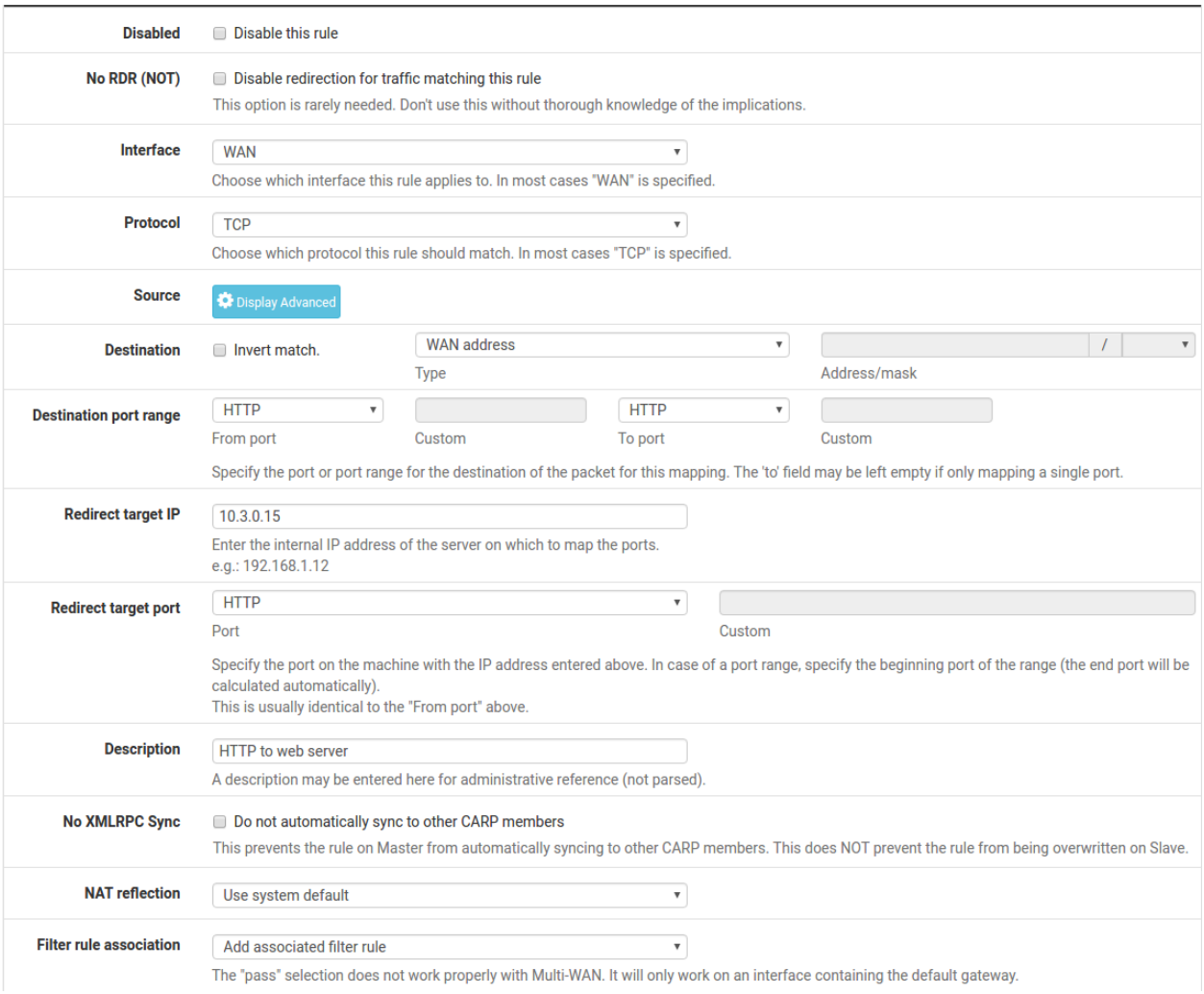


Fig. 1: Port Forward Example

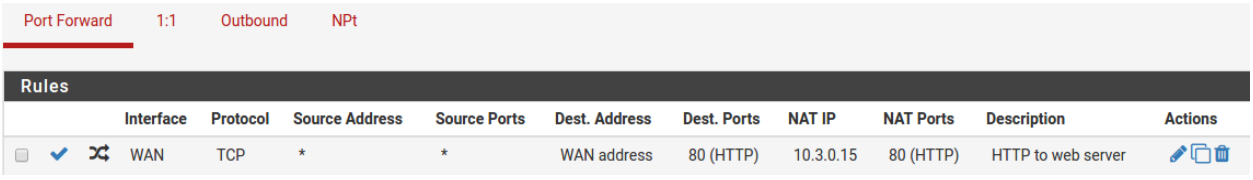


Fig. 2: Port Forward List

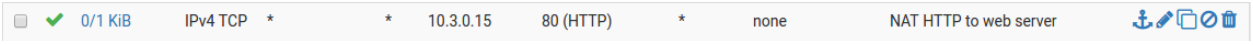


Fig. 3: Port Forward Firewall Rule

14.1.7 Port Forward Limitations

A single port can only be forwarded to one internal host for each available public IP address. For instance, if only one public IP address is available, one internal web server that uses TCP port `80` to serve web traffic can be configured. Any additional servers must use alternate ports such as `8080`. If five available public IP addresses are configured as Virtual IP addresses, then five internal web servers using port `80` can be configured. See *Virtual IP Addresses* for more about Virtual IP addresses.

Tip: For services such as HTTP and HTTPS, port sharing may be possible by using the *HAProxy package*. If the requests can be distinguished in some way, such as by different request hostnames, a proxy can make more advanced decisions about how to forward requests to internal hosts.

There is one uncommon but sometimes applicable exception to this rule. If a particular port must be forwarded to a specific internal host only for certain source IP addresses, and that same port can be forwarded to a different host for other source IP addresses, that is possible by specifying the source address in the port forward entries, such as in Figure *Port Forward Example with Different Sources*.

Rules

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions
	WAN	TCP	bob	*	WAN address	22 (SSH)	10.3.0.5	22 (SSH)	Redirect SSH from Bob to Bob's server	
	WAN	TCP	sue	*	WAN address	22 (SSH)	10.3.0.15	22 (SSH)	Redirect SSH from Sue to Sue's server	

Fig. 4: Port Forward Example with Different Sources

In order for port forwards on WAN addresses to be accessible by using their respective WAN IP address from internal-facing interfaces, NAT reflection must be enabled, which is described in *NAT Reflection*. **Always test port forwards from a system on a different Internet connection**, and not from inside the network. Testing from a mobile device on 3G/4G is a quick and easy way to confirm external connectivity.

14.1.8 Service Self-Configuration With UPnP or NAT-PMP

Some programs support Universal Plug-and-Play (UPnP) or NAT Port Mapping Protocol (NAT-PMP) to automatically configure NAT port forwards and firewall rules. Even more security concerns apply there, but in home use the benefits often outweigh any potential concerns. See *UPnP & NAT-PMP* for more information on configuring and using UPnP and NAT-PMP.

14.1.9 Traffic Redirection with Port Forwards

Another use of port forwards is for transparently redirecting traffic from an internal network. Port forwards specifying the *LAN* interface or another internal interface will redirect traffic matching the forward to the specified destination. This is most commonly used for transparently proxying HTTP traffic to a proxy server, or redirecting all outbound DNS to one server.

The NAT entries shown in Figure *Example Redirect Port Forward* are an example of a configuration that will redirect all HTTP traffic coming into the LAN interface to Squid (port `3128`) on the host `10.3.0.10`, but will not redirect the traffic coming from the squid proxy host itself. They must be in the correct order in the list of port forwards: The negate rule first, then the redirect.

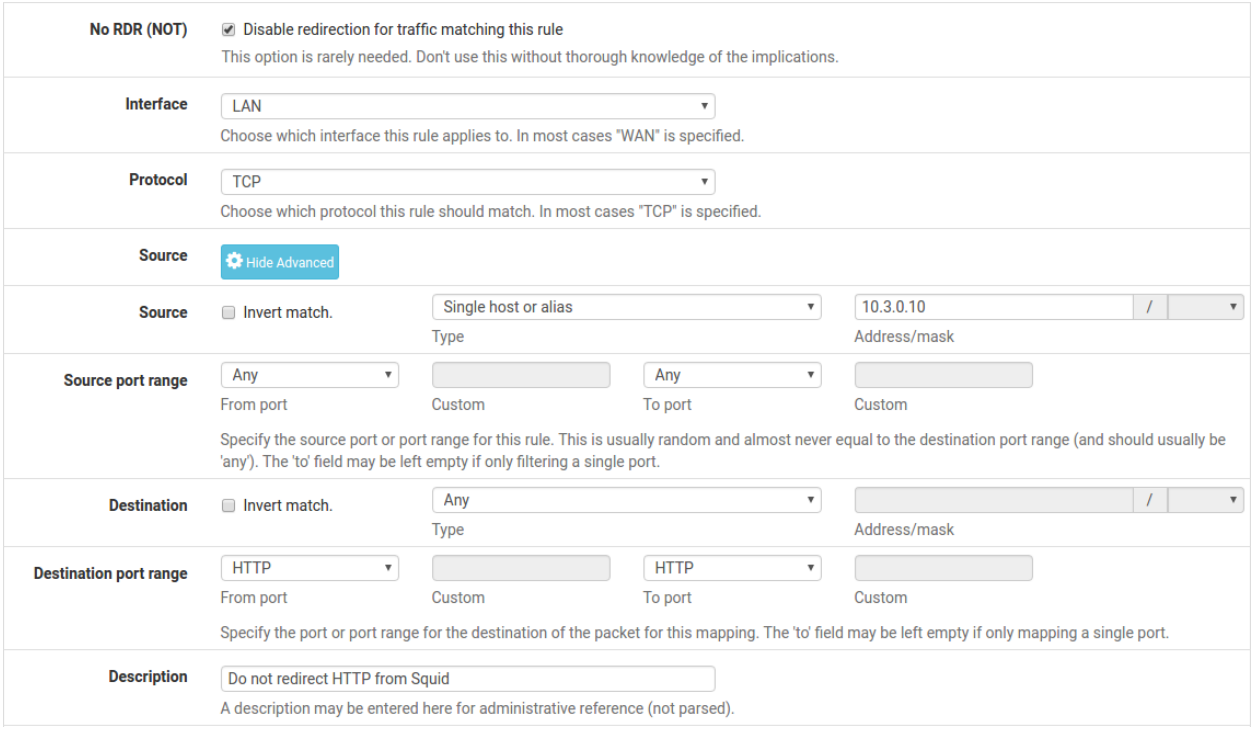


Fig. 5: Example Redirect Port Forward (Negation)

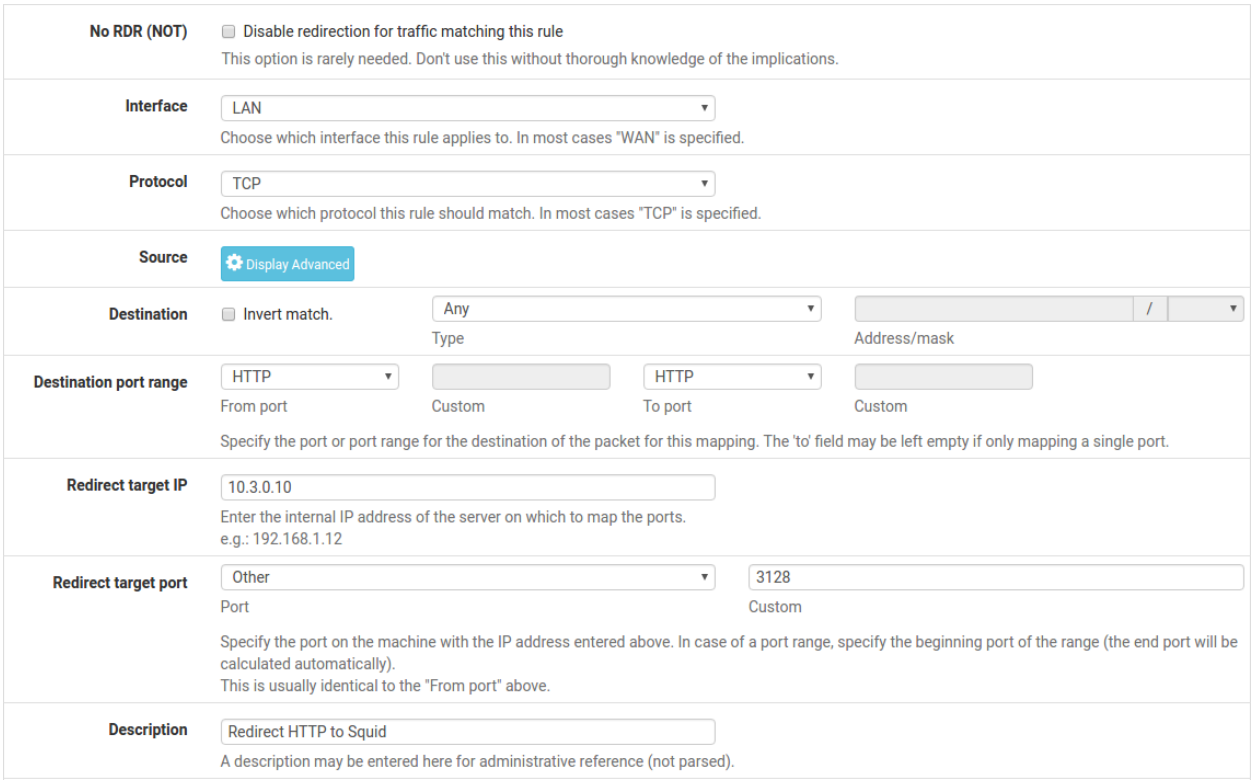


Fig. 6: Example Redirect Port Forward

14.2 1:1 NAT

1:1 NAT (pronounced "one-to-one NAT") maps one external IPv4 address (usually public) to one internal IPv4 address (usually private). All traffic originating from that private IPv4 address going to the Internet will be mapped by 1:1 NAT to the public IPv4 address defined in the entry, overriding the Outbound NAT configuration. All traffic initiated on the Internet destined for the specified public IPv4 address on the mapping will be translated to the private IPv4 address, then evaluated against the WAN firewall ruleset. If matching traffic is permitted by the firewall rules to a target of the private IPv4 address, it will be passed to the internal host.

1:1 NAT can also translate whole subnets as well as single addresses, provided they are of the same size and align on proper subnet boundaries.

The ports on a connection remain constant with 1:1 NAT; For outbound connections, the source ports used by the local system are preserved, similar to using **Static Port** on outbound NAT rules.

14.2.1 Risks of 1:1 NAT

The risks of 1:1 NAT are largely the same as port forwards, if WAN firewall rules permit traffic. Any time rules permit traffic, potentially harmful traffic may be admitted into the local network. There is a slight added risk when using 1:1 NAT in that firewall rule mistakes can have more dire consequences. With port forward entries, traffic is limited by constraints within the NAT rule and the firewall rule. If TCP port 80 is opened by a port forward rule, then an allow all rule on WAN would still only permit TCP 80 on that internal host. If 1:1 NAT rules are in place and an allow all rule exists on WAN, everything on that internal host will be accessible from the Internet. Misconfigurations are always a potential hazard, and this usually should not be considered a reason to avoid 1:1 NAT. Keep this fact in mind when configuring firewall rules, and as always, avoid permitting anything that is not required.

14.2.2 Configuring 1:1 NAT

To configure 1:1 NAT:

- Add a Virtual IP for the public IP address to be used for the 1:1 NAT entry as described in *Virtual IP Addresses*
- Navigate to **Firewall > NAT**, **1:1** tab
- Click  **Add** to create a new 1:1 entry at the top of the list
- Configure the 1:1 NAT entry as follows:

  Disabled Controls whether this 1:1 NAT entry is active.

  Interface The interface where the 1:1 NAT translation will take place, typically a WAN type interface.

  External subnet IP The IPv4 address to which the **Internal IP** address will be translated as it enters or leaves the **Interface**. This is typically an IPv4 Virtual IP address on **Interface**, or an IP address routed to the firewall via **Interface**.

  Internal IP The IPv4 address behind the firewall that will be translated to the **External subnet IP** address. This is typically an IPv4 address behind this firewall. The device with this address must use this firewall as its gateway directly (attached) or indirectly (via static route). Specifying a subnet mask here will translate the entire network matching the subnet mask. For example using `x.x.x.0/24` will translate anything in that subnet to its equivalent in the external subnet.

  Destination Optional, a network restriction that limits the 1:1 NAT entry. When a value is present, the 1:1 NAT will only take effect when traffic is going from the **Internal IP** address to the

Destination address on the way out, or from the **Destination** address to the **External subnet IP** address on the way into the firewall. The Destination field supports the use of aliases.

Description An optional text description to explain the purpose of this entry.

NAT reflection An override for the global NAT reflection options. *Use system default* will respect the global NAT reflection settings, *enable* will always perform NAT reflection for this entry, and *disable* will never do NAT reflection for this entry. For more information on NAT Reflection, see *NAT Reflection*.

- Click **Save**
- Click **Apply Changes**

Example Single IP Address 1:1 Configuration

This section demonstrates how to configure a 1:1 NAT entry with a single internal and external IP address. In this example, `198.51.100.210` is a Virtual IP address on the WAN interface. In most deployments this will be substituted with a working public IP addresses. The mail server in this mapping resides on a DMZ segment using internal IP address `10.3.1.15`. The 1:1 NAT entry to map 198.51.100.210 to 10.3.1.15 is shown in Figure *1:1 NAT Entry*.

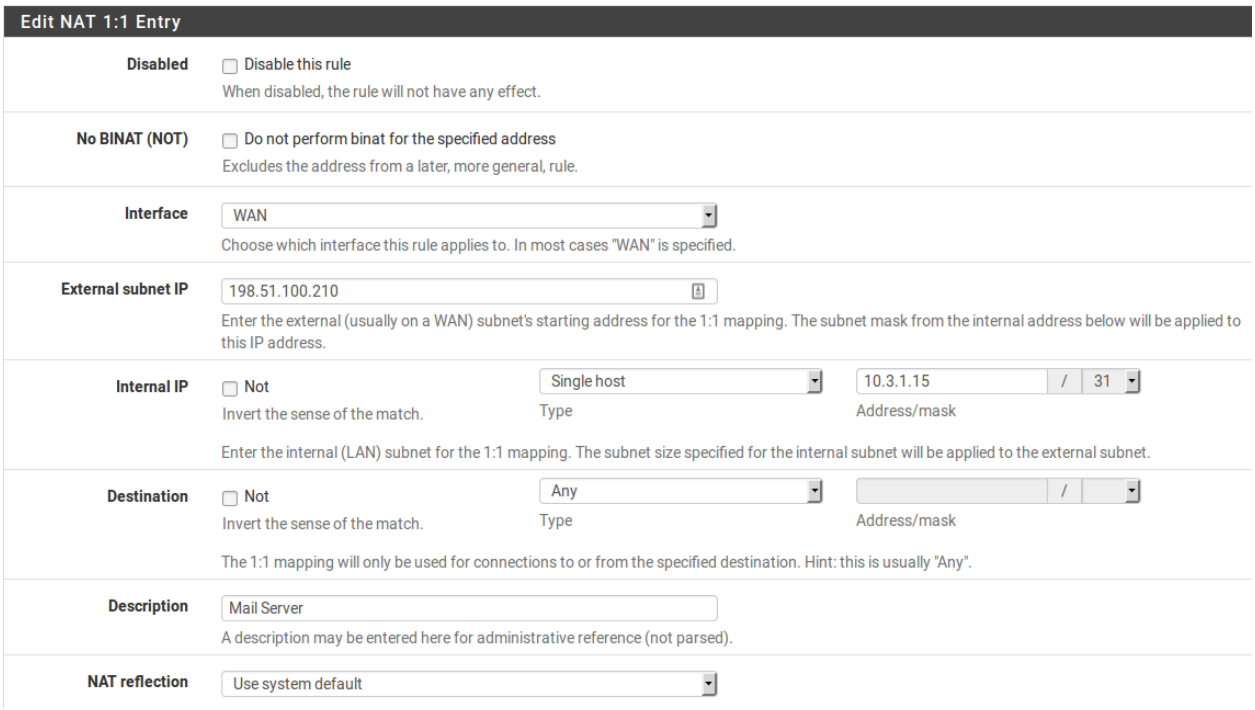


Fig. 7: 1:1 NAT Entry

Example IP Address Range 1:1 Configuration

1:1 NAT can be configured for multiple public IP addresses by using CIDR ranges. In this example, 1:1 NAT is configured for a /30 CIDR range of IPs.

See also:

See *CIDR Summarization* for more information on summarizing networks or groups of IP addresses inside a larger subnet using CIDR notation.

Table 1: /30 CIDR Mapping Matching Final Octet

External IP	Internal IP
198.51.100.64/30	10.3.1.64/30
198.51.100.64	10.3.1.64
198.51.100.65	10.3.1.65
198.51.100.66	10.3.1.66
198.51.100.67	10.3.1.67

The last octet of the IP addresses need not be the same on the inside and outside, but doing so makes it logically simpler to follow. For example, Table */30 CIDR Mapping Non-Matching Final Octet* is also valid.

Table 2: /30 CIDR Mapping Non-Matching Final Octet

External IP	Internal IP
198.51.100.64/30	10.3.1.200/30
198.51.100.64	10.3.1.200
198.51.100.65	10.3.1.201
198.51.100.66	10.3.1.202
198.51.100.67	10.3.1.203

Choosing an addressing scheme where the last octet matches makes the layout easier to understand and hence maintain. Figure *1:1 NAT entry for /30 CIDR range* shows how to configure 1:1 NAT to achieve the mapping listed in Table */30 CIDR Mapping Matching Final Octet*.

14.2.3 1:1 NAT on the WAN IP, aka "DMZ" on Linksys

Some consumer routers such as those from Cisco/Linksys have what they call a "DMZ" feature that will forward all ports and protocols destined to the WAN IP address to a system on the LAN. In effect, this is 1:1 NAT between the WAN IP address and the IP address of the internal system. "DMZ" in that context, however, has nothing to do with what an actual DMZ network is in real networking terminology. In fact, it's almost the opposite. A host in a true DMZ is in an isolated network away from the other LAN hosts, secured away from the Internet and LAN hosts alike. In contrast, a "DMZ" host in the Linksys meaning is not only on the same network as the LAN hosts, but completely exposed to incoming traffic with no protection.

In pfSense® software, 1:1 NAT can be active on the WAN IP address, with the caveat that it will leave all services running on the firewall itself inaccessible externally. So 1:1 NAT cannot be used on the WAN IP address in cases where VPNs of any type are enabled, or other local services on the firewall must be accessible externally. In some cases, this limitation can be mitigated by a port forward for locally hosted services.

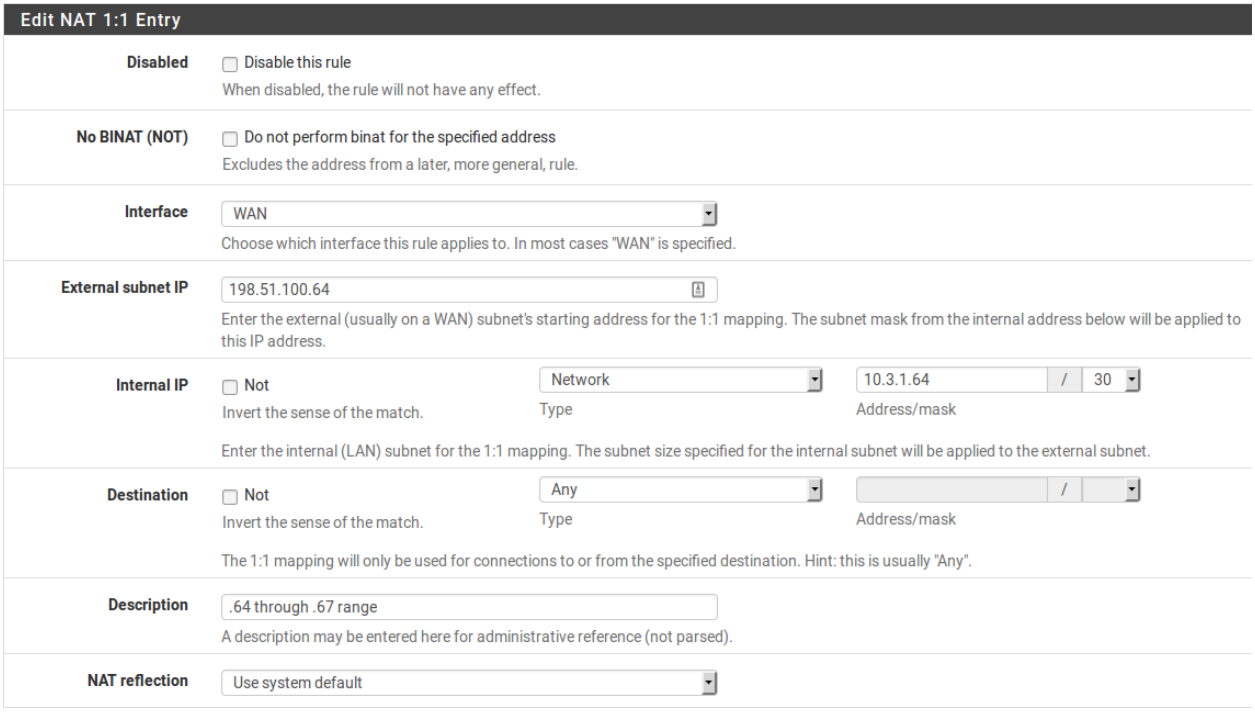


Fig. 8: 1:1 NAT entry for /30 CIDR range

14.3 Ordering of NAT and Firewall Processing

Understanding the order in which firewalling and NAT occurs is important when configuring NAT and firewall rules. The basic logical order is illustrated by Figure *Ordering of NAT and Firewall Processing*. The figure also depicts where tcpdump ties in, since its use as a troubleshooting tool is described later in this documentation in *Packet Capturing*.

Each layer is not always hit in typical configurations, but the use of floating rules or manual outbound NAT or other more complicated configurations can hit each layer in both directions. The diagram only covers basic scenarios for inbound and outbound traffic.

In terms of how the ruleset is processed, the order is:

- Outbound NAT rules
- Inbound NAT rules such as Port Forwards (including `rdr pass` and UPnP)
- Rules dynamically received from RADIUS for IPsec and OpenVPN clients
- Internal automatic rules (pass and block for various items like lockout, snort, DHCP, etc.)
- User-defined rules:
  - Rules defined on the *floating tab*
  - Rules defined on interface group tabs (Including IPsec and OpenVPN)
  - Rules defined on interface tabs (WAN, LAN, OPTx, etc)
- Automatic VPN rules

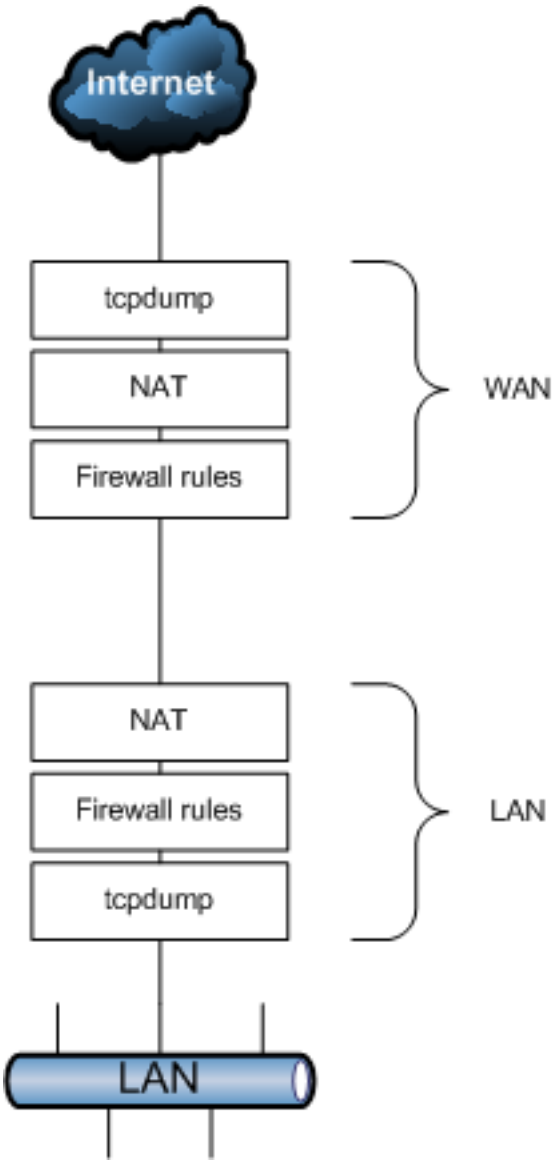


Fig. 9: Ordering of NAT and Firewall Processing

14.3.1 Firewall/NAT Processing Order Example

Traffic from LAN to WAN is processed as described in the following more detailed example. If a type of rules do not exist or do not match, they are skipped.

- Port forwards or 1:1 NAT on the LAN interface (e.g. proxy or DNS redirects)
- Firewall rules for the LAN interface:
  - Floating rules **inbound** on LAN
  - Rules for interface groups including the LAN interface
  - LAN tab rules
- 1:1 NAT or Outbound NAT rules on WAN
- Floating rules that match **outbound** on WAN

In this case, port forwards on WAN and WAN tab firewall rules do not apply.

For traffic initiated on the WAN, the order is the same but direction is reversed:

- Port forwards or 1:1 NAT on the WAN interface (e.g. public services)
- Firewall rules for the WAN interface:
  - Floating rules **inbound** on WAN
  - Rules for interface groups including the WAN interface
  - WAN tab rules
- 1:1 NAT or Outbound NAT rules on LAN
- Floating rules that match **outbound** on LAN

`tcpdump` is always the first and last thing to see traffic, depending on the direction. First, on the incoming interface before any NAT and firewall processing, and last on the outbound interface. It shows what is on the wire. (See *Packet Capturing*)

See also:

See *Rule Processing Order* for more information about the firewall rule processing order.

14.3.2 Floating Rules notes

Floating rules without `quick` set process as "last match wins" instead of "first match wins". Therefore, if a floating rule is set without `quick` and a packet matches that rule, then it also matches a later rule, the later rule will be used. This is the opposite of the other tab rules (groups, interfaces) and rules with `quick` set which stop processing as soon as a match is made. See *Floating Rules* for more details on how floating rules operate.

Extrapolating to additional interfaces

The previous diagram and lists only illustrate a basic two interface LAN and WAN deployment. When working with additional interfaces, the same rules apply. Traffic between two internal interfaces behaves the same as LAN to WAN traffic, though the default NAT rules will not translate traffic between internal interfaces so the NAT layer does not do anything in those cases. If Outbound NAT rules exist that match traffic between internal interfaces, it will apply as shown.

Rules for NAT

On the way into an interface, NAT applies before firewall rules, so if the destination is translated on the way in (e.g. port forwards or 1:1 NAT on WAN), then the firewall rules must match the translated destination. In the typical case of a port forward on WAN, this means the rule must match a destination of the target private IP address on LAN.

For example, with a port forward for TCP port 80 on WAN with an automatically added firewall rule, Figure *Firewall Rule for Port Forward to LAN Host* shows the resulting firewall rule on WAN. The internal IP address on the port forward is 10.3.0.15. Whether using port forwards or 1:1 NAT, firewall rules on all WAN interfaces must use the internal IP address as the destination.

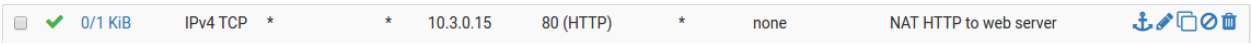


Fig. 10: Firewall Rule for Port Forward to LAN Host

On the way out of an interface, outbound NAT applies before firewall rules, so any floating rules matching outbound on an interface must match the source after it has been translated by outbound NAT or 1:1 NAT.

14.4 NAT Reflection

NAT reflection refers to the ability to access external services from the internal network using the external (usually public) IP address, the same as if the client were on the Internet. While many commercial and open source firewalls do not support this functionality at all, pfSense® software has good support for NAT reflection, though some environments will require a split DNS infrastructure to accommodate this functionality.

When possible, split DNS is the preferred means of accessing resources so that the firewall is not involved in accessing internal services internally. Split DNS is covered at the end of this section in *Split DNS*.

14.4.1 Configuring NAT Reflection

To enable NAT Reflection globally:

- Navigate to **System > Advanced** on the **Firewall & NAT**
- Locate the **Network Address Translation** section of the page
- Configure the NAT Reflection options as follows:

NAT Reflection mode for Port Forwards There are three available choices for NAT Reflection mode for port forwards, they are:

Disable NAT Reflection will not be performed, but it may be enabled on a per-rule basis.

NAT + Proxy Enables NAT Reflection using a helper program to send packets to the target of the port forward. This is useful in setups where the interface and/or gateway IP address used for communication with the target cannot be accurately determined at the

time the rules are loaded. Reflection rules for use with the proxy are not created for ranges larger than 500 ports and will not be used for more than 1000 ports total between all port forwards. This mode does not work with UDP, only with TCP. Because this is a proxy, the source address of the traffic, as seen by the server, is the firewall IP address closest to the server.

Pure NAT Enables NAT Reflection using only NAT rules in pf to direct packets to the target of the port forward. It has better scalability, but it must be possible to accurately determine the interface and gateway IP address used for communication with the target at the time the rules are loaded. There are no inherent limits to the number of ports other than the limits of the protocols. All protocols available for port forwards are supported. If servers are on the same subnet as clients, the **Enable automatic outbound NAT for Reflection** option will mask the source of the traffic so it flows properly back through the firewall.

Reflection Timeout This option is only relevant to *NAT + Proxy* mode, and controls how long the NAT proxy daemon will wait before closing a connection. Specify the value in seconds.

Enable NAT Reflection for 1:1 NAT This option allows clients on internal networks to reach locally hosted services by connecting to the external IP address of a 1:1 NAT entry. To fully activate the feature, check both **Enable NAT Reflection for 1:1 NAT** and **Enable automatic outbound NAT for Reflection**. The latter option is only necessary if clients and servers are in the same subnet.

Enable automatic outbound NAT for Reflection When enabled, this option activates additional NAT rules for 1:1 NAT Reflection and Pure NAT mode NAT Reflection for port forwards. These additional rules mask the source address of the client to ensure reply traffic flows back through the firewall. Without this, connections between the client and server will fail as the server will reply directly back to the client using its internal IP address. The client will drop the connection since it expects a reply from the public IP address.

- Click **Save** to activate the new NAT reflection options

NAT Reflection Caveats

NAT reflection is a hack as it loops traffic through the firewall when it is not necessary. Because of the limited options pf allows for accommodating these scenarios, there are some limitations in the pfSense NAT + Proxy reflection implementation. Port ranges larger than 500 ports do not have NAT reflection enabled in NAT + Proxy mode, and that mode is also effectively limited to only working with TCP. The other modes require additional NAT to happen if the clients and servers are connected to the same interface of the firewall. This extra NAT hides the source address of the client, making the traffic appear to originate from the firewall instead, so that the connection can be properly established.

Split DNS is the best means of accommodating large port ranges and 1:1 NAT. Maintaining a split DNS infrastructure is required by many commercial firewalls even, and typically isn't a problem.

14.4.2 Split DNS

A preferable alternative to NAT reflection is deploying a split DNS infrastructure. Split DNS refers to a DNS configuration where, for a given hostname, public Internet DNS resolves to public IP address, and DNS on the internal network resolves to the internal, private IP address. The means of accommodating this will vary depending on the specifics of an organization's DNS infrastructure, but the end result is the same. NAT reflection is not necessary because hostnames resolve to the private IP addresses inside the network and clients can reach the servers directly.

Split DNS allows servers to see the true client IP address, and connections between servers and clients in the same subnet will go directly, rather than unnecessarily involving the firewall.

The only case that does not work properly with split DNS is when the external and internal port numbers are different. With split DNS, the port number has to be the same in both places.

DNS Resolver/Forwarder Overrides

If pfSense is acting as the DNS server for internal hosts, then host overrides in the DNS Resolver or DNS forwarder can provide split DNS functionality.

To add an override to the DNS Resolver:

- Navigate to **Services > DNS Resolver**
- Click  the under **Host Overrides** to reach the Host Override Options page
- Configure the host override as needed, using the internal IP address of the server. See *Host Overrides*. Figure *Add DNS Resolver Override for example.com* shows an example of a DNS override for `example.com` and `www.example.com`.
- Click **Save**
- Click **Apply Changes**

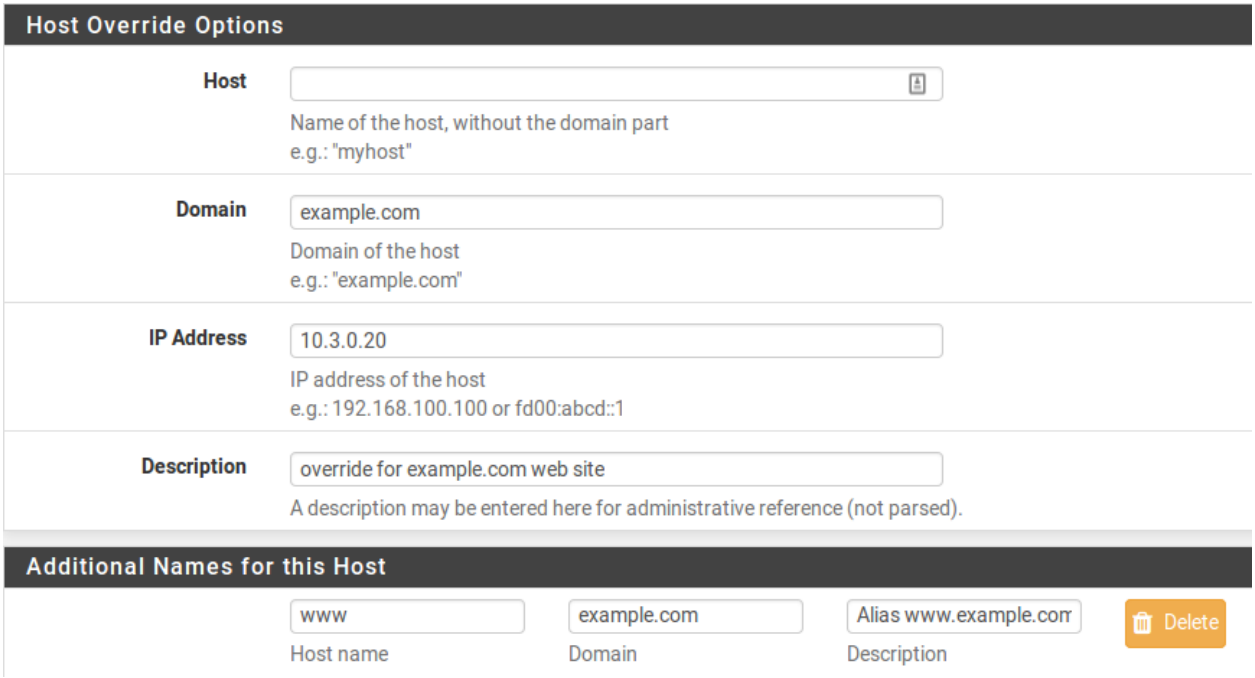


Fig. 11: Add DNS Resolver Override for example.com

The DNS Forwarder works identically in this regard. If the DNS Forwarder is enabled instead of the DNS Resolver, add the overrides there.

An override is required for each hostname in use behind the firewall.

Internal DNS servers

When using a separate DNS server on an internal network, such as Microsoft Active Directory, zones must be created by the DNS server administrator for all domains hosted inside the network, along with all other records for those domains (A, CNAME, MX, etc.).

In environments running the BIND DNS server where the public DNS is hosted on the same server as the private DNS, BIND's views feature is used to resolve DNS differently for internal hosts than external ones. Other DNS servers may support similar functionality. Check their documentation for information.

14.5 Outbound NAT

Outbound NAT, also known as *Source NAT*, controls how pfSense® will translate the source address and ports of traffic leaving an interface. To configure Outbound NAT, navigate to **Firewall > NAT**, on the **Outbound** tab.

There are four possible **Modes** for Outbound NAT:

Automatic Outbound NAT The default option, which automatically performs NAT from internal interfaces, such as LAN, to external interfaces, such as WAN.

Hybrid Outbound NAT Utilizes manual rules while also using automatic rules for traffic not matched by manually entered rules. This mode is the most flexible and easy to use for administrators who need a little extra control but do not want to manage the entire list manually.

Manual Outbound NAT Only honors the manually entered rules, and nothing more. Offers the most control, but can be tough to manage and any changes made to internal interfaces or WANs must be accounted for in the rules by hand. If the list is empty when switching from automatic to manual, the list is populated with rules equivalent to the automatically generated set.

Disable Outbound NAT Disables all outbound NAT. Useful if the firewall contains only routable addresses (e.g. public IP addresses) on all LANs and WANs.

When changing the **Mode** value, click the **Save** button to store the new value.

In networks with a single public IP address per WAN, there is usually no reason to enable manual outbound NAT. If some manual control is necessary, hybrid mode is the best choice. In environments with multiple public IP addresses and complex NAT requirements, manual outbound NAT offers more fine-grained control over all aspects of translation.

For environments using High Availability with CARP, it is important to NAT outbound traffic to a CARP VIP address, as discussed in *High Availability*. This can be accomplished in either hybrid or manual mode.

As with other rules in pfSense, outbound NAT rules are considered from the top of the list down, and the first match is used. Even if rules are present in the Outbound NAT screen, they will not be honored unless the **Mode** is set to **Hybrid Outbound NAT** or **Manual Outbound NAT**.

Note: Outbound NAT only controls what happens to traffic *as it leaves an interface*. It does *not* control the interface though which traffic will exit the firewall. That is handled by the routing table (*Static Routes*) or policy routing (*Policy routing*).

14.5.1 Default Outbound NAT Rules

When set to the default **Automatic Outbound NAT** mode, pfSense maintains a set of NAT rules to translate traffic leaving any internal network to the IP address of the WAN interface which the traffic leaves. Static route networks and remote access VPN networks are also included in the automatic NAT rules.

When outbound NAT is configured for **Automatic** or **Hybrid** modes, the automatic rules are presented in the lower section of the screen labeled **Automatic Rules**.

If the Outbound NAT rule list is empty, switching to **Manual Outbound NAT** and saving will generate a full set of rules equivalent to the automatic rules.

14.5.2 Static Port

By default, pfSense rewrites the source port on all outgoing connections except for UDP port 500 (IKE for VPN traffic). Some operating systems do a poor job of source port randomization, if they do it at all. This makes IP address spoofing easier and makes it possible to fingerprint hosts behind the firewall from their outbound traffic. Rewriting the source port eliminates these potential (but unlikely) security vulnerabilities. Outbound NAT rules, including the automatic rules, will show  in the **Static Port** column on rules set to randomize the source port.

Source port randomization breaks some rare applications. The default Automatic Outbound NAT ruleset disables source port randomization for UDP 500 because it will almost always be broken by rewriting the source port. Outbound NAT rules which preserve the original source port are called **Static Port** rules and have  on the rule in the **Static Port** column. All other traffic has the source port rewritten by default.

Other protocols, such as those used by game consoles, may not work properly when the source port is rewritten. To disable this functionality, use the **Static Port** option.

To add a rule for a device which requires static source ports:

- Navigate to **Firewall > NAT**, **Outbound** tab
- Select **Hybrid Outbound NAT** rule generation
- Click **Save**
- Click  to add a new NAT rule to the top of the list
- Configure the rule to match the traffic that requires static port, such as a source address of a PBX or a game console (See *Working with Manual Outbound NAT Rules* below)
- Check **Static Port** in the **Translation** section of the page
- Click **Save**
- Click **Apply Changes**

After making that change, the source port on outgoing traffic matching the rule will be preserved. The best practice is to use strict rules when utilizing static port to avoid any potential conflict if two local hosts use the same source port to talk to the same remote server and port using the same external IP address.

14.5.3 Disabling Outbound NAT

If public IP addresses are used on local interfaces, and thus NAT is not required to pass traffic through the firewall, disable NAT for the routable subnet. This can be achieved in several ways:

- If NAT is not required for any interface, set the outbound NAT mode to **Disable**
- Using Hybrid Outbound NAT, a rule set with **Do not NAT** can disable NAT for matching traffic
- Using Manual Outbound NAT, delete (or do not create) any NAT rules matching the routable subnets

In any of the above cases, outbound NAT will no longer be active for those source IP addresses and pfSense will then route public IP addresses without translation.

14.5.4 Working with Manual Outbound NAT Rules

Outbound NAT rules are very flexible and are capable of translating traffic in many ways.

The NAT rules are shown in a single page and the **Interface** column is a source of confusion for some; As traffic leaves an interface, only the outbound NAT rules set for that specific **Interface** are consulted.

Click  from the Outbound NAT page to add a rule to the **top** of the list. Click  to add a rule to the bottom. Place specific rules at the top, and more general rules at the bottom. The rules are processed by the firewall starting at the top of the list and working down, and the first rule to match is used. Rules may be reordered to match in the desired way.

The options for each Outbound NAT rule are:

Disabled Toggles whether or not this rule is active.

Do not NAT Checking this option causes packets matching the rule to *not* have NAT applied as they leave. This is necessary if the traffic would otherwise match a NAT rule, but must not have NAT applied. One common use for this is to add a rule exception so that the firewall IP addresses do not get NAT applied, especially in the case of CARP, where such NAT would break Internet communication from a secondary node while it is in backup mode.

Interface The interface where this NAT rule will apply when traffic is leaving via this interface. Typically this is WAN or an OPT WAN, but in some special cases it could be LAN or another internal interface.

Protocol In most cases, Outbound NAT will apply to *any* protocol, but occasionally it is necessary to restrict the protocol upon which the NAT will act. For example, to only perform static port NAT for UDP traffic from a PBX.

Source The Source is the local network which will have its address translated as it leaves the selected **Interface**. This is typically a LAN, DMZ, or VPN subnet. The Source Port is nearly always left blank to match all ports. This field supports the use of aliases if the **Type** is set to *Network*.

Note: Avoid using a source address of *any* as that will also match traffic from the firewall itself. This will cause problems with gateway monitoring and other firewall-initiated traffic.

Destination In most cases, the **Destination** remains set to *any* so that traffic going anywhere out of this **Interface** will be translated, but the Destination can be restricted as needed. For example, to translate in a certain way when going to a specific destination, such as only doing static port NAT to SIP trunk addresses. This field supports the use of aliases if the **Type** is set to *Network*.

Translation The **Address** field inside of the **Translation** section controls what happens to the source address of traffic matching this rule. Most commonly, this is set to *Interface Address* so the traffic

is translated to the IP address of **Interface**, e.g. the WAN IP address. The **Address** drop-down also contains all defined Virtual IP addresses, host aliases, and *Other Subnet* to manually enter a subnet for translation.

Note: An alias containing subnets cannot be used for translation. Only host aliases or a single manually entered subnet may be used.

Using a host alias or manually entered subnet, an outbound NAT rule can translate to a pool of addresses. This can help in large NAT deployments or in areas where static port is required for several clients. When translating to a host alias or subnet, a **Pool Options** drop-down is available with several options. Only *Round Robin* types work with host aliases. Any type may be used with a subnet.

Default Does not define any specific algorithm for selecting a translation address from the pool.

Round Robin Loops through each potential translation address in the alias or subnet in turn.

Round Robin with Sticky Address Works the same as *Round Robin* but maintains the same translation address for a given source address as long as states from the source host exist.

Random Selects a translation address for use from the subnet at random.

Random with Sticky Address Selects an address at random, but maintains the same translation address for a given source address as long as states from the source host exist.

Source Hash Uses a hash of the source address to determine the translation address, ensuring that the translated address is always the same for a given source IP address.

Bitmask Applies the subnet mask and keeps the last portion identical. For example if the source address is `10.10.10.50` and the translation subnet is `192.2.0.0/24`, the rule will change the address to `192.2.0.50`. This works similarly to 1:1 NAT but only in the outbound direction.

Port Specifies a specific *source* port for translation. This is almost always left blank, but could be required if the client selects a random source port but the server requires a specific source port.

Static Port Causes the original source port of the client traffic to be maintained after the source IP address has been translated. Some protocols require this, like IPsec without NAT-T, and some protocols behave better with this, such as SIP and RTP. Checking this option disables the **Port** entry box.

No XML-RPC Sync This option is only relevant if an HA Cluster configuration is in use, and should be skipped otherwise. When using an HA cluster with configuration synchronization, checking this box will prevent the rule from being synchronized to the other members of a cluster (see *High Availability*). Typically all rules should synchronize, however. This option is only effective on master nodes, it does *not* prevent a rule from being overwritten on slave nodes.

Description An optional text reference to explain the purpose of this rule.

These rules can accommodate most any NAT scenario, large or small.

14.5.5 Tracking Changes to Outbound NAT Rules

As mentioned in Figure *Firewall Rule Time Stamps* for firewall rules, a timestamp is added to an outbound NAT entry indicating when it was created or last edited. This timestamp shows which user created the rule, and the last person to edit the rule. When switching from Automatic Outbound NAT mode to Manual Outbound NAT mode, the created rules are marked as being created by that process.

14.6 Choosing a NAT Configuration

The best NAT configuration for a given deployment depends primarily on the number of public IP addresses available and the number of local services that require inbound access from the Internet.

14.6.1 Single Public IP Address per WAN

When only a single public IP per WAN is available, NAT options are limited. 1:1 NAT rules can be used with WAN IP addresses, but that can have drawbacks. In this case, we recommend only using port forwards.

14.6.2 Multiple Public IP Addresses per WAN

When multiple public IP addresses are available per WAN, numerous options are available for inbound and outbound NAT configuration. Port forwards, 1:1 NAT, and Hybrid or Manual Outbound NAT may all be desirable, depending on the needs of the site.

14.7 NAT and Protocol Compatibility

Some protocols do not work well with NAT and others will not work at all. Problematic protocols embed IP addresses and/or port numbers within packets (e.g. SIP and FTP), some do not work properly if the source port is rewritten (SIP from a PBX, IPsec), and some are difficult because of limitations of pf (PPTP). This section covers a sampling of protocols that have difficulties with NAT in pfSense® software, and how to work around these issues where possible.

14.7.1 FTP

FTP poses problems with both NAT and firewalls because of the design of the protocol. FTP was initially designed in the 1970s, and the current standard defining the specifications of the protocol was written in 1985. Since FTP was created more than a decade prior to NAT, and long before firewalls were common, it acts in ways that are very unfriendly toward NAT and firewalls.

pfSense does not include an FTP proxy by default, but there is a client proxy available as an add-on package.

FTP Limitations

Because pf lacks the ability to properly handle FTP traffic without a proxy, and the FTP proxy package implementation is somewhat lacking, there are some restrictions on the usage of FTP.

FTP servers behind NAT

For FTP servers behind NAT, all relevant ports must be manually forwarded in to the server and allowed in firewall rules. Or in the case of 1:1 NAT, only the firewall rules are necessary. Depending on the FTP mode, server software, and client software, some server configuration may also be required.

FTP modes

FTP can act in multiple modes that change the behavior of the client and server, and which side listens for incoming connections. The complications of NAT and firewall rules depend on these modes and whether a remote client is attempting to reach a server behind pfSense, or if a client behind pfSense is attempting to reach a remote server.

Active Mode

With Active Mode FTP, when a file transfer is requested, the client listens on a local port and then tells the server the client IP address and port. The server will then connect back to that IP address and port in order to transfer the data. This is a problem for firewalls because the port is typically random, though modern clients allow for limiting the range that is used. In the case of a client behind NAT, the IP address given would be a local address, unreachable from the server. Not only that, but a firewall rule would need to be added along with a port forward allowing traffic into this port.

When the FTP proxy package is in use and a client is behind pfSense connecting to a remote server, the proxy attempts to do three major things: First, it will rewrite the FTP PORT command so that the IP address is the WAN IP address of the firewall, and a randomly chosen port on that IP address. Next, it adds a port forward that connects the translated IP address and port to the original IP address and port specified by the FTP client. Finally, it allows traffic from the FTP server to connect to that "public" port. With Multi-WAN, the proxy will only function on the WAN containing the default gateway.

When everything is working properly, this all happens transparently. The server never knows it is talking to a client behind NAT, and the client never knows that the server isn't connecting directly.

In the case of a server behind NAT, active mode is not usually a problem since the server will only be listening for connections on the standard FTP ports and then making outbound connections back to the clients. The outbound firewall rules must allow the server to make arbitrary outbound connections, and the rules must not policy route those connections out a WAN other than the one that accepted the inbound FTP connection.

Passive Mode

Passive Mode (PASV) acts somewhat in reverse. For clients, it is more NAT and firewall friendly because the server listens on a port when a file transfer is requested, not the client. Typically, PASV mode will work for FTP clients behind NAT without using any proxy or special handling at all.

Similar to the situation in the previous section, when a client requests PASV mode the server will provide the client with its IP address and a random port to which the client can attempt to connect. Since the server is on a private network, that IP address and port will need to be translated and allowed through the firewall. See *FTP Servers and Port Forwards* below for rule requirements. The FTP server must provide the public IP address to which clients

connect, but some clients such as Filezilla are smart enough to ignore a given IP address if it is private, and will connect to the original server IP address instead.

Extended Passive Mode

Extended Passive Mode (EPSV) works similar to PASV mode but makes allowances for use on IPv6. When a client requests a transfer, the server will reply with the port to which the client should connect. The same caveats for servers in PASV mode apply here.

FTP Servers and Port Forwards

For FTP servers providing passive mode to clients, the configuration of the FTP server must define a passive port range and must also set the external NAT address, typically the WAN IP address of the firewall. The means of setting these values varies depending on the FTP server software implementation. Consult the FTP server documentation for more information. On the firewall, the passive port range must be forwarded in with port forwards along with TCP port 21.

For FTP servers providing active mode to clients, a port forward is only required for TCP port 21.

FTP Servers and 1:1 NAT

With 1:1 NAT, firewall rules must allow port 21 and the passive port range.

14.7.2 TFTP

Standard TCP and UDP traffic initiates connections to remote hosts using a random source port in the ephemeral port range, which varies by operating system but falls within 1024-65535, and the destination port of the protocol in use. Replies from server to client reverse that: The source port is the client destination port, and the destination port is the client source port. This is how pf associates the reply traffic with connections initiated from inside a network.

TFTP (Trivial File Transfer Protocol) does not follow this convention, however. The standard defining TFTP, RFC 1350, specifies the reply from the TFTP server to client will be sourced from a pseudo-random port number. The TFTP client may choose a source port of 10325 (as an example) and use the destination port for TFTP, port 69. The server for other protocols would then send the reply using source port 69 and destination port 10325. Since TFTP instead uses a pseudo-random source port, the reply traffic will not match the state pf has created for this traffic. Hence the replies will be blocked because they appear to be unsolicited traffic from the Internet.

TFTP is not a commonly used protocol across the Internet. The only situation that occasionally comes up where this is an issue is with some IP phones that connect to outside VoIP providers on the Internet using TFTP to pull configuration and other information. Most VoIP providers do not require this.

If TFTP traffic must pass through the firewall, a TFTP proxy is available which is configured under **System > Advanced** on the **Firewall & NAT** tab. See *TFTP Proxy* for more information.

14.7.3 PPTP / GRE

The limitations with PPTP in pfSense are caused by limitations in the ability of pf to NAT the GRE protocol. As such, the limitations apply to any use of the GRE protocol, however PPTP has been the most common use of GRE in the wild.

The state tracking code in pf for the GRE protocol can only track a single session per public IP address per external server. This means if a PPTP VPN connection is in place, only one internal machine can connect simultaneously to the same a PPTP server on the Internet. A thousand machines can connect simultaneously to a thousand different PPTP servers, but only one simultaneously to a single server. A single client can also connect to an unlimited number of outside PPTP servers.

The only available work around is to use multiple public IP addresses on the firewall, one per client via Outbound or 1:1 NAT, or to use multiple public IP addresses on the external PPTP server. This is not a problem with other types of VPN connections.

Due to the extremely flawed security in PPTP (See *PPTP Warning*), including a complete compromise of the entire protocol, its usage should be discontinued as soon as possible, so this issue is not relevant given the current security standards.

14.7.4 Online Games

Games typically are NAT friendly aside from a couple caveats. This section refers to both PC games and console gaming systems with online capabilities. This section provides an overview of the experiences of numerous pfSense users. Visit the Gaming board on the pfSense forum to find more information.

Static Port

Some games do not work properly unless static port is enabled on outbound NAT rules. If a game has problems establishing a connection, the best thing to try first is enabling static port for traffic coming from the console. See *Static Port* for more information.

Multiple players or devices behind one NAT device

Some games have issues where multiple players or devices are behind a single NAT device. These issues appear to be specific to NAT, not pfSense, as users who have tried other firewalls experience the same problems with them as well. Search the Gaming board on the pfSense forum for the game or system to find information from others with similar experiences.

Overcome NAT issues with UPnP

Many modern game systems support Universal Plug-and-Play (UPnP) to automatically configure any required NAT port forwards and firewall rules. Enabling UPnP on pfSense will typically allow games to work with little or no intervention. See *UPnP & NAT-PMP* for more information on configuring and using UPnP, and for information on potential security concerns.

14.8 IPv6 Network Prefix Translation (NPt)

Network Prefix Translation, or NPt for short, works similarly to 1:1 NAT but operates on IPv6 prefixes instead. NPt can be found under **Firewall > NAT** on the **NPt** tab.

NPt takes one prefix and translates it to another. So `2001:db8:1111:2222::/64` becomes `2001:db8:3333:4444::/64` and though the prefix changes, the remainder of the address will be identical for a given host on that subnet.

Warning: NPt **does NOT** function like traditional outbound/overload NAT/PAT. NPt cannot be used to map an internal prefix to prefix or single address in use on a WAN, it must be used with a routed prefix.

There are a few purposes for NPt, but many question its actual usefulness. With NPt, "private" IPv6 space (`fc00::/7`) can be utilized on a LAN and it can be translated by NPt to a public, routed, IPv6 prefix as it comes and goes through a WAN. The utility of this is debatable. One use is to avoid renumbering the LAN if external providers change, however since anything external that looked for the old prefix must also be adjusted, the usefulness of that can go either way, especially when the configuration must account for avoiding collisions in the `fc00::/7` space for VPN tunnels.

NPt makes perfect sense for SOHO IPv6 Multi-WAN deployments. The likelihood that a home or small business end user will have their own provider-independent IPv6 space and a BGP feed is very small. In these cases, the firewall can utilize a routed prefix from multiple WANs to function similarly to Multi-WAN on IPv4. As traffic leaves the second WAN sourced from the LAN subnet, NPt will translate it to the equivalent IP address in the routed subnet for that WAN. The LAN can either use one of the routed prefixes and do NPt on the other WANs, or use addresses in `fc00::/7` and do NPt on all WANs. We recommend avoiding use of the `fc00::/7` space for this task. For more information on Multi-WAN with IPv6, see *Configuring Multi-WAN for IPv6*.

When adding an NPt entry, there are few options to consider as NPt is fairly basic:

Disabled Toggles whether this rule is actively used.

Interface Selects the Interface where this NPt rule takes effect as the traffic exits.

Internal IPv6 Prefix The local (e.g. LAN) IPv6 subnet and prefix length, typically the `/64` on LAN or other internal network.

Destination IPv6 Prefix The routed external IPv6 subnet and prefix length to which the Internal IPv6 Prefix will be translated. This is **NOT** the prefix of the WAN itself. It must be a network routed to this firewall via **Interface**

Description A brief description of the purpose for this entry.

Figure *NPt Example* shows an NPt rule where the LAN IPv6 subnet `2001:db8:1111:2222::/64` will be translated to `2001:db8:3333:4444::/64` as it leaves the HENETV6DSL interface.

See also:

- *Configuring NAT for a VoIP PBX*
- *Configuring NAT for VoIP Phones*
- *Accessing Port Forwards from Local Networks*
- *Using NAT and FTP without a Proxy*
- *Troubleshooting NAT*
- *Troubleshooting NAT Port Forwards*
- *Troubleshooting 1:1 NAT*

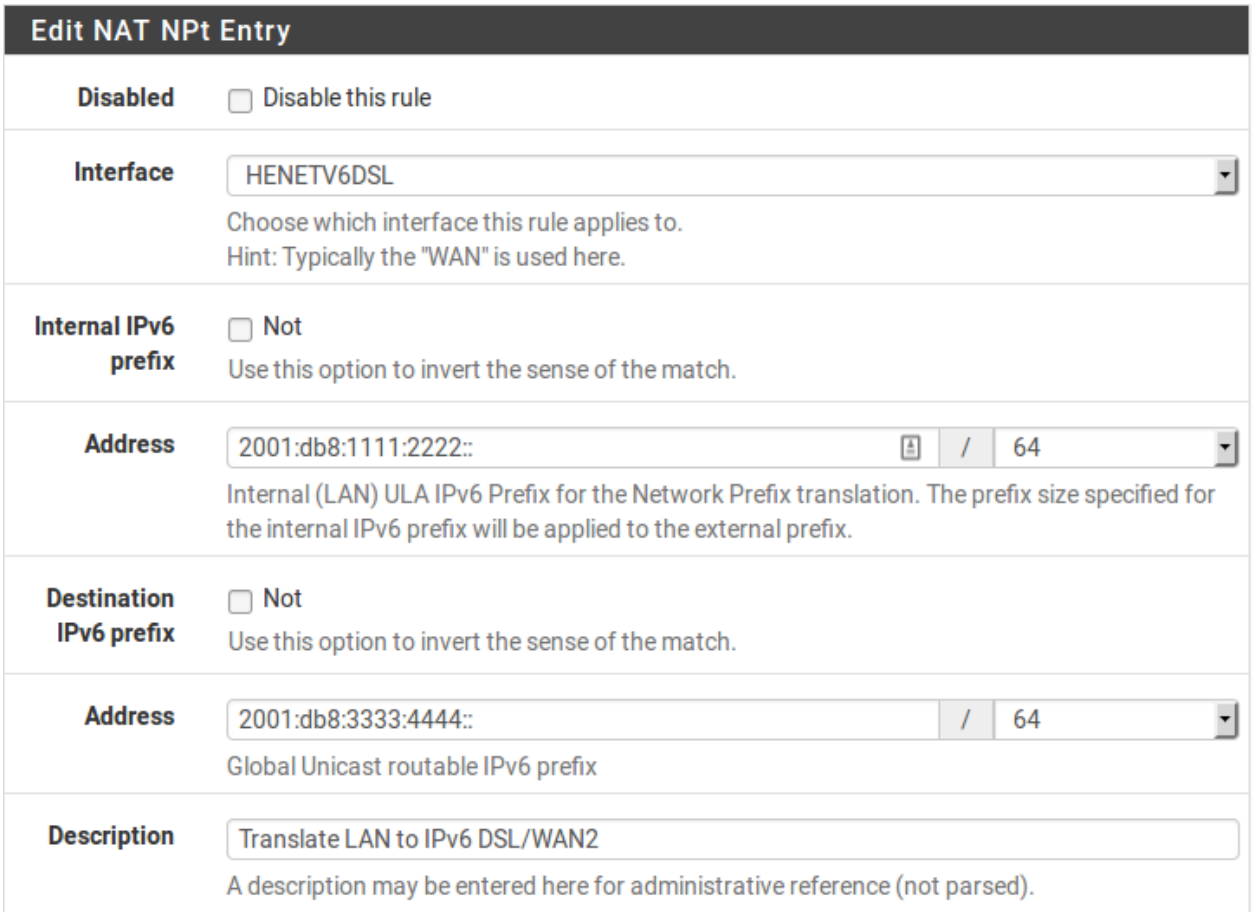


Fig. 12: NPt Example

- *Troubleshooting NAT Reflection*

In its most common usage, Network Address Translation (NAT) allows multiple computers using IPv4 to be connected to the Internet using a single public IPv4 address. pfSense® software enables these simple deployments, but also accommodates much more advanced and complex NAT configurations required in networks with multiple public IP addresses.

NAT is configured in two directions: inbound and outbound. Outbound NAT defines how traffic leaving a local network destined for a remote network, such as the Internet is translated. Inbound NAT refers to traffic entering a network from a remote network. The most common type of inbound NAT is *port forwards*, which is also the type many administrators are most familiar with.

Note: In general, with the exception of Network Prefix Translation (NPt), NAT on IPv6 is not supported in pfSense. There is further discussion on the topic in *IPv6 and NAT*. Unless otherwise mentioned, this chapter is discussing NAT with IPv4.

See also:

pfSense Hangouts on Youtube to view the May 2016 Hangout on NAT with pfSense software version 2.3 and the earlier August 2014 Hangout on Network Address Translation.

14.9 Default NAT Configuration

This section describes the default NAT configuration present on pfSense. The most appropriate NAT configuration that can be determined is generated automatically. In some environments, this configuration may not be suitable, and pfSense fully enables changing it from the web interface. This is a contrast from many other open source firewall distributions, which do not allow the capabilities commonly required in all but small, simple networks.

14.9.1 Default Outbound NAT Configuration

In a typical two-interface pfSense setup with LAN and WAN, the default NAT configuration automatically translates Internet-bound traffic to the WAN IP address. When multiple WAN interfaces are configured, traffic leaving any WAN interface is automatically translated to the address of the WAN interface being used.

Static port is automatically configured for IKE (part of IPsec). Static port is covered in more detail in *Outbound NAT* about Outbound NAT.

For detecting WAN-type interfaces for use with NAT, pfSense looks for the presence of a gateway selected on the interface configuration if it has a static IP address, or pfSense assumes the interface is a WAN if it is a dynamic type such as PPPoE or DHCP.

14.9.2 Default Inbound NAT Configuration

By default, nothing is allowed in from the Internet on the WAN interface. If traffic initiated on the Internet must be allowed to reach a host on the internal network, port forwards or 1:1 NAT are required. This is covered in the coming sections.

CHAPTER

FIFTEEN

ROUTING

15.1 Gateways

Gateways are the key to routing; They are routers through which other networks can be reached. The kind of gateway most people are familiar with is a *default* gateway, which is the router through which a host will communicate to the Internet or any other networks it doesn't have a more specific route to reach. Gateways are also used for static routing, where other networks must be reached via specific local routers. On most networks, gateways reside in the same subnet as one of the interfaces on a host. For example, if a firewall has an IP address of `192.168.22.5/24`, then a gateway to another network would have to be somewhere inside of `192.168.22.x` if the other network is reachable through that interface. One notable exception to this is point-to-point interfaces like those used in PPP-based protocols, which often have gateway IP addresses in another subnet because they are not used in the same way.

15.1.1 Gateway Address Families (IPv4 and IPv6)

When working with routing and gateways, the functionality and procedures are the same for both IPv4 and IPv6 addresses, however all of the addresses for a given route must involve addresses of the same family. For example, an IPv6 network must be routed using an IPv6 gateway/router. A route cannot be created for an IPv6 network using an IPv4 gateway address. When working with gateway groups, the same restriction applies; All gateways in a gateway group must be of the same address family.

15.1.2 Managing Gateways

Before a gateway can be utilized for any purpose, it must be added to the firewall configuration.

If a gateway will be used for a WAN-type interface, it can be added on the configuration page for that interface (See *Interface Configuration Basics*), or it may be added first manually and then selected from the drop-down list on the interface configuration.

Dynamic interface types such as DHCP and PPPoE receive an automatic gateway that is noted as **Dynamic** in the gateway list. The parameters for such gateways can be adjusted the same as the parameters for a static gateway.

Note: Deleting a dynamic gateway will clear its custom settings, but the dynamic gateway itself cannot be removed.

To add or manage gateways, navigate to **System > Routing**, **Gateways** tab.

On the screen there are a variety of options to manage gateway entries:

-  **Add** at the bottom of the list creates a new gateway

-  creates a copy of an existing gateway
-  edits an existing gateway
-  deletes a gateway
-  disables an active gateway
-  enables a disabled gateway

The individual options for gateways are discussed in detail in *Gateway Settings*.

Managing the Default Gateway

The **Default Gateway** section at the bottom of **System > Routing**, **Gateways** tab controls which gateway(s) are used by default when the firewall routes traffic. Traffic from the firewall itself will follow the default gateway, as will traffic passing through the firewall when it does not match other more specific routes or policy routing rules.

There are two controls in the section which set the default gateway for IPv4 and IPv6 respectively.

The default gateway can have one of the following values:

Automatic The firewall will automatically use gateways from this list (from the top down) for the default gateway, switching to the next item in the list if gateways fail or are marked down.

For more control over this behavior, use a gateway group instead.

Gateway The selected single gateway is always used for the default gateway.

Gateway Group The firewall uses the selected *gateway group* pick the default gateway. It will change from one gateway to another if the preferred default fails.

> **Warning:** This function does not support load balancing, only failover. When using a gateway group for the default gateway, the group must only have one gateway in each tier.

None No default gateway for the address family will be added to the routing table.

15.2 Gateway Settings

When adding or editing a gateway, the GUI presents a page with the options for controlling gateway behavior.

The only required settings are the **Interface**, **Address Family**, **Name**, and the **Gateway** (IP address).

Interface The interface through which the gateway is reached. For example, if this is a local gateway on the LAN subnet, choose the *LAN* interface here.

Address Family Either *IPv4* or *IPv6*, depending on the type of address for this gateway.

Name The **Name** for the gateway, as referenced in the gateway list, and various drop-down and other selectors for gateways. It can only contain alphanumeric characters, or an underscore, but no spaces. For example: `WANGW`, `GW_WAN`, and `WANGATE` are valid but `WAN GW` is not allowed.

Gateway The IP address of the gateway. As mentioned previously, this must reside in a subnet directly configured on the selected **Interface**.

Note: Rare cases which require a gateway not in the interface subnet can still function, but require an additional setting. See the **Use Non-Local Gateway** option in *Advanced Gateway Settings* for details.

Disable Gateway Monitoring By default, the gateway monitoring daemon will ping each gateway periodically to monitor latency and packet loss for traffic to the monitored IP address. This data is used for gateway status information and also to draw the Quality RRD graph. If this monitoring is undesirable for any reason, it may be disabled by checking **Disable Gateway Monitoring**. Note that if the gateway status is not monitored, then Multi-WAN will not work properly as it cannot detect failures.

Disable Gateway Monitoring Action When set, the gateway monitoring daemon will take no action if the status of the gateway changes. For example, no events will be acted upon if it becomes unresponsive or suffers from high latency.

This is useful if the administrator wants to monitor a gateway without the monitoring causing additional disruptions.

Monitor IP The **Monitor IP** address option configures the IP address used by the gateway monitoring daemon to determine the gateway status using ICMP echo requests ("pings").

By default the gateway monitoring daemon will ping the gateway IP address. This is not always desirable, especially in the case where the gateway IP address is local, such as on a cable modem or fiber CPE. In those cases it makes more sense to ping something farther upstream, such as an ISP DNS server or a server on the Internet. Another case is when an ISP is prone to upstream failures, so pinging a host on the Internet is a more accurate test to determine if a WAN is usable rather than testing the link itself. Some popular choices include Google public DNS servers, or popular web sites such as Google or Yahoo. If the IP address specified in this box is not directly connected, a static route is added to ensure that traffic to the **Monitor IP** address leaves via the expected gateway. Each gateway must have a unique Monitor IP address.

The status of a gateway as perceived by the firewall can be checked by visiting **Status > Gateways** or by using the **Gateways** widget on the dashboard. If the gateway shows **Online**, then the monitor IP address is successfully responding to pings.

Force State When **Mark Gateway as Down** is checked, the gateway will always be considered down, even when pings are returned from the monitor IP address. This is useful for cases when a WAN is behaving inconsistently and the gateway transitions are causing disruption. The gateway can be forced into a *down* state so that other gateways may be preferred until it stabilizes.

Description An optional **Description** of the gateway entry for reference. A short note about the gateway or interface it's used for may be helpful, or it may be left blank.

15.2.1 Advanced Gateway Settings

Several parameters can be changed to control how a gateway is monitored or treated in a Multi-WAN scenario. Most users will not need to alter these values. To access the advanced options, click the  **Display Advanced** button. If any of the advanced options are set, this section is automatically expanded. For more information on using multiple WAN connections, see *Multiple WAN Connections*.

Weight When using Multi-WAN load balancing, if two gateways have different amounts of bandwidth, the **Weight** parameter adjusts the ratio at which connections are sent through each gateway.

For example if WAN1 has 50Mbit/s and WAN2 has 100Mbit/s, weight WAN1 as *1* and WAN2 as *2*. Then for every three connections that go out, one will use WAN1 and two will use WAN2. Using this method, connections are distributed in a way that is more likely to better utilize the available bandwidth. Weight from *1* to *30* may be chosen.

Data Payload To conserve bandwidth, the `dpinger` daemon sends a ping with a payload size of `0` by default so that no data is contained within the ICMP echo request. However, in rare circumstances a CPE, ISP router, or intermediate hop may drop or reject ICMP packets without a payload. In these cases, set the payload size above `0`. Usually a size of `1` is enough to satisfy affected equipment.

Latency Thresholds The **Latency Thresholds** fields control the amount of latency that is considered normal for this gateway. This value is expressed in milliseconds (ms). The default values are **From** `200` and **To** `500`.

The value in the **From** field is the lower boundary at which the gateway would be considered in a warning state, but not down. If the latency exceeds the value in the **To** field, it is considered down and removed from service. The proper values in these fields can vary depending on what type of connection is in use, and what ISP or equipment is between the firewall and the monitor IP address.

Some common situations may require adjusting these values. For instance some DSL lines operate acceptably even at higher latency, so increasing the **To** parameter to `700` or more would lower the number of times the gateway would be considered down when, in fact, it was passing traffic sufficiently. Another example is a GIF tunnel to a provider such as he.net for IPv6. Due to the nature of GIF tunnels and load on the tunnel servers, the tunnel could be working acceptably even with latency as high as `900` ms as reported by ICMP ping responses.

Packet Loss Thresholds Similar to **Latency Thresholds**, the **Packet Loss Thresholds** control the amount of packet loss to a monitor IP address before it would be considered unusable. This value is expressed as a percentage, `0` being no loss and `100` being total loss. The default values are **From** `10` and **To** `20`.

The value in the **From** field is the lower boundary at which the gateway would be considered in a warning state, but not down. If the amount of packet loss exceeds the value in the **To** field, it is considered down and removed from service. The proper values in these fields can vary depending on what type of connection is in use, and what ISP or equipment is between the firewall and the monitor IP address.

As with latency, connections can be prone to different amounts of packet loss and still function in a usable way, especially if the path to a monitor IP address drops or delays ICMP in favor of other traffic. We have observed unusable connections with minor amounts of loss, and some that are usable even when showing 45% loss. If loss alarms occur on a normally functioning WAN gateway, enter higher values in the **From** and **To** fields until a good balance for the circuit is achieved.

Probe Interval The value in the **Probe Interval** field controls how often a ping is sent to the monitor IP address, in *milliseconds*. The default is to ping twice per second (`500` ms).

In some situations, such as links that need monitored but have high data charges, even a small ping every second can add up. This value can be safely increased so long as it less than or equal to the

Alert Interval and also does not violate the constraint on the **Time Period** listed below. Lower values will ping more often and be more accurate, but consume more resources. Higher values will be less sensitive to erratic behavior and consume less resources, at the cost of accuracy.

Note: The quality graph is averaged over seconds, not intervals, so as the **Probe Interval** is increased the accuracy of the quality graph is decreased.

Loss Interval Time in milliseconds before packets are treated as lost. The default is `2000` ms (2 seconds). Must be greater than or equal to the **High Latency Threshold**.

If a circuit is known to have high latency while operating normally, this can be increased to compensate.

Time Period The amount of time, in milliseconds, over which ping results are averaged. The default is `60000` (60 seconds, one minute). A longer **Time Period** will take more time for latency or loss to trigger an alarm, but it is less prone to be affected by erratic behavior in ping results.

The **Time Period** must be greater than **twice** the sum of the **Probe Interval** and **Loss Interval**, otherwise there may not be at least one completed probe.

Alert Interval The time interval, in milliseconds, at which the daemon checks for an alert condition. The default value is `1000` (1 second). This value must be greater than or equal to the **Probe Interval**, because an alert could not possibly occur between probes.

Use Non-Local Gateway The **Use non-local gateway through interface specific route** option allows a non-standard configuration where a gateway IP address exists outside of an interface subnet. Some providers attempting to scrape the bottom of the IPv4 barrel have resorted to this in order to not put a gateway into each customer subnet. Do not activate this option unless required to do so by the upstream provider.

15.3 Gateway Groups

Gateway groups are a set of gateways, but are treated as one entity in gateway fields of the GUI. Groups will appear in the gateway drop-downs available on, for example, firewall rule editing.

Gateway groups are managed from the **Groups** tab on **System > Routing**.

15.3.1 Gateway Group Options

When creating a gateway group, the following options are available:

Group Name The name of this gateway group. The name must be less than 32 characters in length, and may only contain letters a-z, digits 0-9, and an underscore. This will be the name used to refer to this gateway group in the **Gateway** field in firewall rules. This field is required.

Gateway Priority This list contains every gateway on the firewall to select which gateways will be a part of this group. The GUI will filter the list address family after the first selection.

Tier The priority level for this gateway. The value may be from 1-5 or *Never* to exclude the gateway from this group.

Lower values are higher priority. For example, gateways on *Tier 1* are used before gateways on *Tier 2*, and so on.

Gateways on the same tier are used by the firewall for load balancing when possible. Load balancing naturally performs failover as failed gateways are removed from the pool available for load balancing.

Gateways on different tiers result in failover from gateways on lower tiers to those higher tiers. For example, if Tier 1 contains only one gateway and it fails, then the next tier (Tier 2) is checked for available gateways and the firewall uses those instead, and so on.

> **Warning:** Some firewall features which support gateway groups only support failover, not load balancing. For example, when using a gateway group for the default gateway or as a VPN endpoint, each gateway must be on a separate tier.

Virtual IP When using a gateway group for failover in certain contexts which require binding a specific address, such as IPsec, this option controls which address on an interface is used for that purpose. For example, in an HA pair this could be a CARP VIP used as an endpoint for IPsec tunnels.

Leave it set to the default *Interface Address* when a specific address is not required by any use of the gateway group.

Trigger Level

Configures how the firewall manages the gateway group entries when certain types of gateway events occur.

Member Down Marks the gateway as down only when it is completely down, past one or both of the higher thresholds configured for the gateway. This catches the worst sort of failures, when the gateway is completely unresponsive, but may miss more subtle issues with the circuit that can make it unusable long before the gateway reaches that level.

Packet Loss Marks the gateway as down when packet loss crosses the lower alert threshold (See *Advanced Gateway Settings*).

High Latency Marks the gateway as down when latency crosses the lower alert threshold (See *Advanced Gateway Settings*).

Packet Loss or High Latency Marks the gateway as down for either type of alert.

Description Text describing the purpose of this gateway group.

15.3.2 Tier Priority Example

Example:

- WANGW: Tier1
- OPT1GW: Tier2
- OPT3GW: Tier3

In the example above OPT1GW would be used if WANGW fails, OPT3GW will be used if both WANGW and OPT1GW fail.

15.3.3 Connection-Based Round-Robin Load Balancing Example

Example:

- WANGW: Tier1
- OPT1GW: Tier1
- OPT3GW: Tier1

In the example above all gateways have the same **Tier** value. When this group is used by a firewall rule, connections matching that rule will perform connection-based round-robin load balancing between all of the gateways.

Note: If any of the gateways fail, they are automatically removed from active usage in the group, effectively resulting in failover in addition to load balancing.

15.3.4 See Also

Multi-WAN Guide

15.4 Static Routes

Static routes are used when hosts or networks are reachable through a router other than the default gateway. The firewall knows about the networks directly attached to it, and it reaches all other networks as directed by the routing table. In networks where an internal router connects additional internal subnets, a static route must be defined for those networks to be reachable.

Note: The routers through which these other networks are reached must first be added as gateways. See *Gateways* for information on adding gateways.

Static routes are managed at **System > Routing** on the **Routes** tab.

See also:

- *Accessing Firewall Services over IPsec VPNs*
- *Policy Routing Configuration*

15.4.1 Static Route Configuration

When adding or editing a static route, the following options are available:

Destination Network The network and subnet mask reachable using this route. This may be an IPv4 address (subnet ID), IPv6 prefix, or an *alias*.

> **Warning:** Using an alias for a route destination has severe limitations. Routes do not automatically follow changes to aliases. When adding entries to an existing alias, this route must be re-saved and re-applied. When removing entries from an alias, there is no automatic way to clear the routes. Manually remove them from the routing table at the CLI or reboot the firewall.

Gateway The router through which this network is reached.

Disabled Check if the static route should not be used, only defined.

Description Text to describe the route, its purpose, etc.

15.4.2 Managing Static Routes

To add a route:

- Navigate to **System > Routing** on the **Routes** tab
- Click **Add** to create a new static route
- Fill in the configuration as described in *Static Route Configuration*
- Click **Save**
- Click **Apply Changes**

To manage existing routes, navigate to **System > Routing** on the **Routes** tab. On the screen there are a variety of options to manage routes:

- edits an existing route
- creates a copy of an existing gateway
- deletes a route
- disables an active route
- enables a disabled route

15.4.3 Example Static Route

Figure *Static Routes* illustrates a scenario where a static route is required.

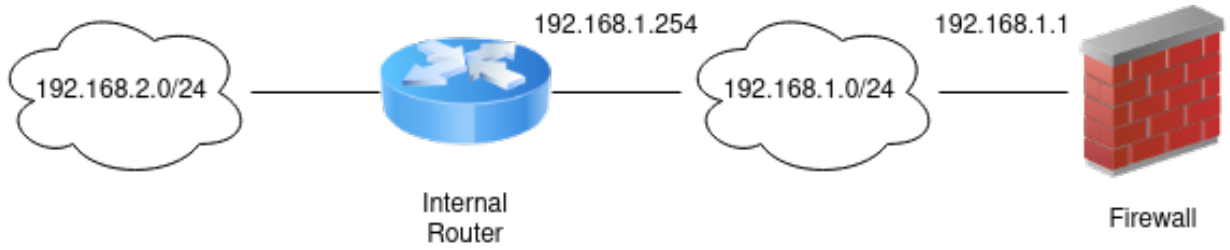


Fig. 1: Static Routes

Because the `192.168.2.0/24` network in Figure *Static Routes* is not on an interface directly connected to the firewall, a static route is required for the firewall to know how to reach that network. Figure *Static Route Configuration* shows the appropriate static route for the above diagram. As mentioned earlier, before a static route may be added a gateway must first be defined.

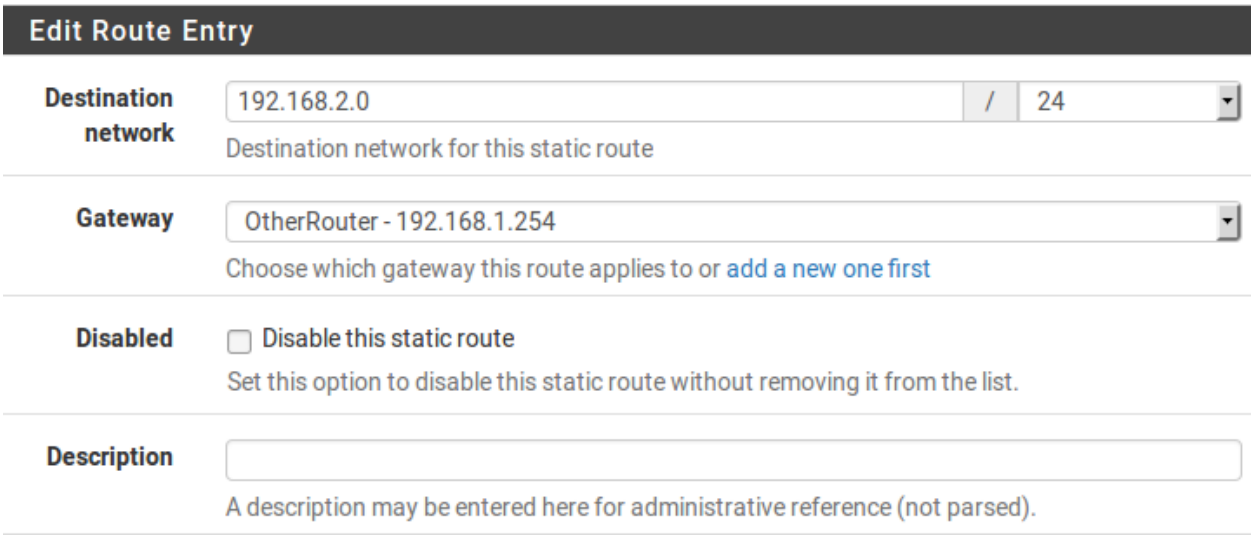


Fig. 2: Static Route Configuration

LAN firewall rules must allow traffic to pass from a source of the networks reachable via static routes on LAN, and outbound NAT must also accommodate these networks.

15.4.4 Bypass Firewall Rules for Traffic on Same Interface

In some environments using static routes, traffic ends up routing asymmetrically. This means the traffic will follow a different path in one direction than the traffic flowing in the opposite direction. Take Figure *Asymmetric Routing* for example.

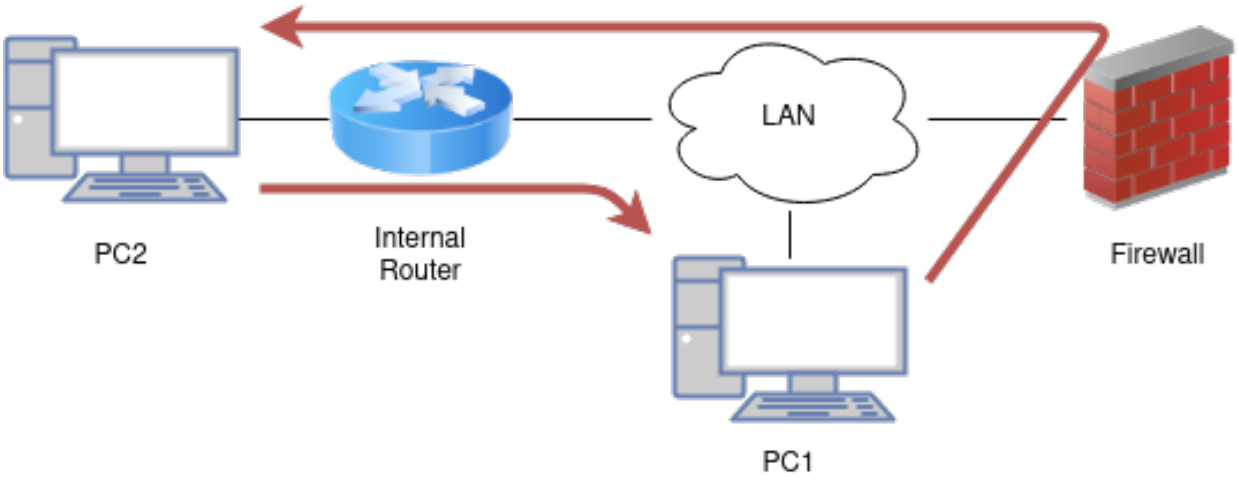


Fig. 3: Asymmetric Routing

Traffic from PC1 to PC2 will go through the firewall since it is the default gateway for PC1, but traffic in the opposite direction will go directly from the router to PC1.

Since this is a stateful firewall, it must see traffic for the entire connection to be able to filter traffic properly. With asymmetric routing such as in this example, any stateful firewall will drop legitimate traffic because it cannot properly keep state without seeing traffic in both directions. This generally only affects TCP, since other protocols do not have

a formal connection handshake the firewall can recognize for use in state tracking.

Note: A connection may appear to work for a short time and then stop. This can be from the firewall removing a state which doesn't transition to a fully open state, or it may be from clients expiring an ICMP redirect.

In asymmetric routing scenarios, there is an option in the firewall GUI which can be used to prevent legitimate traffic from being dropped. The option adds firewall rules which allow all traffic between networks defined in static routes using a more permissive set of rule options and state handling. To activate this option:

- Click **System > Advanced**
- Click the **Firewall/NAT** tab
- Check **Bypass firewall rules for traffic on the same interface**
- Click **Save**

Alternatively, firewall rules may be added manually to allow similar traffic. Two rules are needed, one on the interface tab where the traffic enters (e.g. LAN) and another on the **Floating** tab:

- Navigate to **Firewall > Rules**
- Click the tab for the interface where the traffic will enter (e.g. **LAN**)
- Click  **Add** to add a new rule to the top of the list
- Use the following settings:

  Protocol *TCP*

  Source The local systems utilizing the static route (e.g. *LAN Net*)

  Destination The network on the other end of the route

  TCP Flags Check **Any flags** (Under **Advanced Features**)

  State Type *Sloppy State* (Under **Advanced Features**)
- Click **Save**
- Click the **Floating** tab
- Click  **Add** to add a new rule to the top of the list
- Use the following settings:

  Interface The interface where the traffic originated (e.g. **LAN**)

  Direction *Out*

  Protocol *TCP*

  Source The local systems utilizing the static route (e.g. *LAN Net*)

  Destination The network on the other end of the route

  TCP Flags Check **Any flags** (Under **Advanced Features**)

  State Type *Sloppy State* (Under **Advanced Features**)
- Click **Save**

If additional traffic from other sources or destinations is shown as blocked in the firewall logs with TCP flags such as "TCP:SA" or "TCP:PA", the rules may be adjusted or copied to match that traffic as well.

Note: If filtering of traffic between statically routed subnets is required, it must be done on the router and not the firewall since the firewall is not in a position on the network where it can effectively control that traffic.

See also:

- *Troubleshooting Asymmetric Routing*

15.4.5 ICMP Redirects

When a device sends a packet to its default gateway, and the gateway knows the sender can reach the destination network via a more direct route, it will send an ICMP redirect message in response and forward the packet as configured. The ICMP redirect causes a route for that destination to be temporarily added to the routing table of the sending device, and the device will subsequently use that more direct route to reach that destination.

This will only work if the client OS is configured to permit ICMP redirects, which is typically the case by default.

ICMP redirects are common when static routes are present which point to a router on the same interface as client PCs and other network devices. The asymmetric routing diagram from the previous section is an example of this.

ICMP redirects have a mostly undeserved bad reputation from some in the security community because they allow modification of a client routing table. However they are not the risk that some imply, as to be accepted, the ICMP redirect message must include the first 8 bytes of data from the original datagram. A host in a position to see that data and hence be able to successfully forge illicit ICMP redirects is in a position to accomplish the same end result in multiple other ways.

See also:

- *Route Table Contents*
- *Multiple WAN Connections*
- *IPv6 Router Advertisements*
- *FRR Package*
- *Routing Public IP Addresses*
- *Dynamic Routing Protocol Basics*
- *Troubleshooting Gateway Monitoring*
- *Troubleshooting "No buffer space available" Errors*
- *Troubleshooting Network Connectivity*
- *Troubleshooting Traceroute Output*
- *Troubleshooting Website Access*
- *Troubleshooting Routes*

One of the primary functions of a firewall is routing traffic. This chapter covers several topics related to routing, including gateways, static routes, routing protocols, routing of public IP addresses, and displaying routing information.

CHAPTER

SIXTEEN

BRIDGING

16.1 Creating a Bridge

In pfSense® software, bridges are added and removed at **Interfaces > Assignments** on the **Bridges** tab. Using bridges, any number of ports may be bound together easily. Each bridge created in the GUI will also create a new bridge interface in the operating system, named `bridgeX` where `X` starts at 0 and increases by one for each new bridge. These interfaces may be assigned and used like most other interfaces, which is discussed later in this chapter.

To create a bridge:

- Navigate to **Interfaces > Assignments** on the **Bridges** tab.
- Click **Add** to create a new bridge.
- Select at least one entry from **Member Interfaces**. Select as many as needed using `Ctrl`-click.
- Add a **Description** if desired.
- Click **Show Advanced Options** to review the remaining configuration parameters as needed. For most cases they are unnecessary.
- Click **Save** to complete the bridge.

Note: A bridge may consist of a single member interface, which can help with migrating to a configuration with an assigned bridge, or for making a simple span/mirror port.

16.2 Advanced Bridge Options

There are numerous advanced options for a bridge and its members. Some of these settings are quite involved, so they are discussed individually in this section.

16.2.1 (Rapid) Spanning Tree Options

Spanning Tree is a protocol that helps switches and devices determine if there is a loop and cut it off as needed to prevent the loop from harming the network. There are quite a few options that control how spanning tree behaves which allow for certain assumptions to be made about specific ports or to ensure that certain bridges get priority in the case of a loop or redundant links. More information about STP may be found in the FreeBSD ifconfig(8) man page, and on Wikipedia.

Protocol

The **Protocol** setting controls whether the bridge will use IEEE 802.1D Spanning Tree Protocol (*STP*) or IEEE 802.1w Rapid Spanning Tree Protocol (*RSTP*). RSTP is a newer protocol, and as the name suggests it operates much faster than STP, but is backward compatible. The newer IEEE 802.1D-2004 standard is based on RSTP and makes STP obsolete.

Select STP only when older switch gear is in use that does not behave well with RSTP.

STP Interfaces

The **STP Interfaces** list reflects the bridge members upon which STP is enabled. `Ctrl`-click to select bridge members for use with STP.

Valid Time

Set the **Valid Time** for a Spanning Tree Protocol configuration. The default is `20` seconds. The minimum is `6` seconds and the maximum is `40` seconds.

Forward Time

The **Forward Time** option sets the time that must pass before an interface begins forwarding packets when Spanning Tree is enabled. The default is `15` seconds. The minimum is `4` seconds and the maximum is `30` seconds.

Note: A longer delay will be noticed by directly connected clients as they will not be able to pass traffic, even to obtain an IP address via DHCP, until their interface enters forwarding mode.

Hello Time

The **Hello Time** option sets the time between broadcasting of Spanning Tree Protocol configuration messages. The **Hello Time** may only be changed when operating in legacy STP mode. The default is `2` seconds. The minimum is `1` second and the maximum is `2` seconds.

Bridge Priority

The **Bridge Priority** for Spanning Tree controls whether or not this bridge would be selected first for blocking should a loop be detected. The default is `32768`. The minimum is `0` and the maximum is `61440`. Values must be a multiple of `4096`. Lower priorities are given precedence, and values lower than `32768` indicate eligibility for becoming a root bridge.

Hold Count

The transmit **Hold Count** for Spanning Tree is the number of packets transmitted before being rate limited. The default is `6`. The minimum is `1` and the maximum is `10`.

Port Priorities

The **Priority** fields set the Spanning Tree priority for each bridge member interface. Lower priorities are given preference when deciding which ports to block and which remain forwarding. Default priority is `128`, and must be between `0` and `240`.

Path Costs

The **Path Cost** fields sets the Spanning Tree path cost for each bridge member. The default is calculated from the link speed. To change a previously selected path cost back to automatic, set the cost to `0`. The minimum is `1` and the maximum is `200000000`. Lower cost paths are preferred when making a decision about which ports to block and which remain forwarding.

16.2.2 Cache Settings

Cache Size sets the maximum size of the bridge address cache, similar to the MAC or CAM table on a switch. The default is `100` entries. If there will be a large number of devices communicating across the bridge, set this higher.

Cache entry expire time controls the timeout of address cache entries in seconds. If set to `0`, then address cache entries will not be expired. The default is `240` seconds (Four minutes).

16.2.3 Span Port

Selecting an interface as the **Span port** on the bridge will transmit a copy of every frame received by the bridge to the selected interface. This is most useful for snooping a bridged network passively on another host connected to the span ports of the bridge with something such as Snort, tcpdump, etc. The selected span port may not be a member port on the bridge.

16.2.4 Edge Ports / Automatic Edge Ports

If an interface is set as an **Edge port**, it is always assumed to be connected to an end device, and *never* to a switch; It assumes that the port can never create a layer 2 loop. Only set this on a port when it will never be connected to another switch. By default ports automatically detect edge status, and they can be selected under **Auto Edge** ports to *disable* this automatic edge detection behavior.

16.2.5 PTP Ports / Automatic PTP Ports

If an interface is set as a **PTP port**, it is always assumed to be connected to a switch, and not to an end user device; It assumes that the port can potentially create a layer 2 loop. It should only be enabled on ports that are connected to other RSTP-enabled switches. By default ports automatically detect PTP status, and they can be selected under Auto **PTP ports** to *disable* this automatic PTP detection behavior.

16.2.6 Sticky Ports

An interface selected in **Sticky Ports** will have its dynamically learned addresses cached as though they were static once they enter the cache. Sticky entries are never removed from the address cache, even if they appear on a different interface. This could be used a security measure to ensure that devices cannot move between ports arbitrarily.

16.2.7 Private Ports

An interface marked as a **Private Port** will not communicate with any other port marked as a **Private Port**. This can be used to isolate end users or sections of a network from each other if they are connected to separate bridge ports marked in this way. It works similar to "Private VLANs" or client isolation on a wireless access point.

16.3 Bridging and Interfaces

A bridge interface (e.g. *bridge0*) itself may be assigned as interface. This allows the bridge to act as a normal interface and have an IP address placed upon it rather than a member interface.

Configuring the IP address on the bridge itself is best in nearly all cases. The main reason for this is due to the fact that bridges are dependent on the state of the interface upon which the IP address is assigned. If the IP address for the bridge is configured on a member interface and that interface is down, the whole bridge will be down and no longer passing traffic. The most common case for this is a wireless interface bridged to an Ethernet LAN NIC. If the LAN NIC is unplugged, the wireless would be dead unless the IP address was configured on the bridge interface and not LAN. Another reason is that if limiters must be used for controlling traffic, then there must be an IP address on the bridge interface for them to work properly. Likewise, in order for Captive Portal or a transparent proxy to function on an internal bridge the IP address must be configured on the assigned bridge and not a member interface.

16.3.1 Swapping Interface Assignments

Before getting too far into talking about moving around bridge interface assignments, it must be noted that these changes should be made from a port that is **not** involved in the bridge. For example, if bridging WLAN to LAN, make the change from WAN or another OPT port. Alternately, download a backup of `config.xml` and manually make the changes. Attempting to make changes to a port while managing the firewall from that port will most likely result loss of access to the GUI, leaving the firewall unreachable.

Easy Method: Move settings to the new interface

The easiest, though not the quickest, path in the GUI is to remove the settings from the LAN interface individually (IP address, DHCP, etc) and then activate them on the newly assigned bridge interface.

Quick but Tricky: Reassign the Bridge as LAN

Though this method is a bit trickier than moving the settings, it can be much faster especially in cases where there are lots of firewall rules on LAN or a complex DHCP configuration. In this method, some hoop-jumping is required but ultimately the bridge ends up as the LAN interface, and it retains the LAN IP address, all of the former firewall rules, DHCP, and other interface configuration.

- Assign and configure the bridge members that have not yet been handled. Review the steps below to ensure the interface settings are correct even if the interfaces have already been assigned and configured.
  - Navigate to **Interfaces > Assignments**
  - Choose the interface from the **Available network ports** list
  - Click **Add**
  - Navigate to the new interface configuration page, e.g. **Interfaces > OPT2**
  - Check **Enable**
  - Enter a **Description** such as `WiredLAN2`
  - Set both **IPv4 Configuration Type** and **IPv6 Configuration Type** to *None*
  - Uncheck both **Block private networks** and **Block bogon networks** if checked
  - Click **Save**
  - Click **Apply Changes**
  - Repeat for additional unassigned future bridge members
- Create the new bridge
  - Navigate to **Interfaces > Assignments** on the **Bridges** tab
  - Click **Add** to create a new bridge
  - Enter a **Description**, such as `LAN Bridge`
  - Select all of the new bridge members **EXCEPT** the *LAN* interface in the **Member interfaces** list
  - Click Save
- Change the bridge filtering **System Tunable** to disable member interface filtering
  - Navigate to **System > Advanced**, **System Tunables** tab
  - Locate the entry for **net.link.bridge.pfil_member** or create a new entry if one does not exist, using that name for the **Tunable**
  - Click  to edit an existing entry
  - Enter `0` in the **Value** field
  - Click **Save**
- Navigate to **Interfaces > Assignments**
- Change the assignment of **LAN** to `bridge0`

- Click **Save**
- Assign and configure the old LAN interface as described previously, setting its IP configuration types to *None* and naming it `WiredLAN`
- Edit the bridge and select the newly assigned `WiredLAN` as a bridge member
- Change the bridge filtering **System Tunable** to enable bridge interface filtering
  - Use the procedure described previously, but set **net.link.bridge.pfil_bridge** to `1`

Now the former LAN interface, along with the new bridge members, are all on a common layer 2 with the bridge assigned as LAN along with the other configuration.

Quickest but Most Difficult: Hand Edit config.xml

Hand editing `config.xml` can be very fast for those familiar with the configuration format in XML. This method is easy to get wrong, however, so be sure to have backups and install media nearby in case a mistake is made.

When hand editing `config.xml` to accomplish this task, do as follows:

- Assign the additional bridge members and set their IP configuration types to *None*
- Create the bridge, including *LAN* and *LAN2* and other bridge members
- Assign the bridge (e.g. as OPT2) and enable it, also with an IP configuration type of *None*
- Download a backup of `config.xml` from **Diagnostics > Backup/Restore**
- Open `config.xml` in a text editor that understands UNIX line endings
- Change the *LAN* assignment to `bridge0`
- Change the former *LAN* assignment to what used to be the bridge (e.g. *OPT2*)
- Edit the bridge definition to refer to *OPT2* and not *LAN*
- Save the changes
- Restore the edited `config.xml` from **Diagnostics > Backup/Restore**

The firewall will reboot with the desired setup. Monitor the console to ensure the settings were applied correctly and no errors are encountered during the boot sequence.

16.3.2 Assigned Bridge MAC Addresses and Windows

The MAC address for a bridge is determined randomly when the bridge is created, either at boot time or when a new bridge is created. That means that on each reboot, the MAC address can change. In many cases this does not matter, but Windows Vista, 7, 8, and 10 use the MAC address of the gateway to determine if they are on a specific network. If the MAC changes, the network identity will change and its status as public, private, etc. may need to be corrected. To work around this, enter a MAC address on the assigned bridge interface to spoof it. Then clients will always see the same MAC for the gateway IP address.

16.4 Bridging and firewalling

Filtering with bridged interfaces functions similar to routed interfaces, but there are some configuration choices to alter exactly how the filtering behaves. By default, firewall rules are applied on each member interface of the bridge on an inbound basis, like any other routed interface.

It is possible to decide whether the filtering happens on the bridge member interfaces, or on the bridge interface itself. This is controlled by two values on **System > Advanced** on the **System Tunables** tab, as seen in Figure *Bridge Filtering Tunables*. The **net.link.bridge.pfil_member** tunable controls whether or not the rules will be honored on the bridge member interfaces. By default, this is on (`1`). The **net.link.bridge.pfil_bridge** tunable controls whether or not the rules will be honored on the bridge interface itself. By default, this is off (`0`). At least one of these must be set to `1`.

net.link.bridge.pfil_member	Packet filter on the member interface	1	
net.link.bridge.pfil_bridge	Packet filter on the bridge interface	0	

Fig. 1: Bridge Filtering Tunables

When filtering on the bridge interface itself, traffic will hit the rules as it enters from any member interface. The rules are still considered "inbound" like any other interface rules, but they work more like an interface group since the same rules apply to each member interface.

16.4.1 Firewall Rule Macros

Only one interface of a bridge will have an IP address set, the others will have none. For these interfaces, their firewall macros such as *OPT1 address* and *OPT1 net* are undefined because the interface has no address and thus no subnet.

If filtering is performed on bridge members, keep this fact in mind when crafting rules and explicitly list the subnet or use the macros for the interface where the IP address resides.

16.5 Bridging Two Internal Networks

When bridging two internal networks as described in *Internal Bridges* there are some special considerations to take for certain services on the firewall.

Note: There are additional requirements and restrictions when bridging wireless interfaces because of the way 802.11 functions. See *Bridging and wireless* for more information.

16.5.1 DHCP and Internal Bridges

When bridging one internal network to another, two things need to be done. First, ensure that DHCP is only running on the interface containing the IP address and not the bridge members without an address. Second, an additional firewall rule may be necessary at the top of the rules on the member interfaces to allow DHCP traffic.

Note: This only applies to filtering being performed on member interfaces, not filtering performed on the bridge.

When creating a rule to allow traffic on an interface, normally the source is specified similar to *OPT1 Subnet* so that only traffic from that subnet is allowed out of that segment. With DHCP, that is not enough. Because a client does not

yet have an IP address, a DHCP request is performed as a broadcast. To accommodate these requests, create a rule on the bridge member interfaces with the following settings:

- Navigate to **Firewall > Rules** on the tab for the bridge member
- Click  **Add** to add a new rule to the top of the list
- **Protocol**: *UDP*
- **Source**: `0.0.0.0`
- **Source Port**: `68`
- **Destination**: `255.255.255.255`
- **Destination port**: `67`
- **Description** stating this will `Allow DHCP`
- Click **Save** and **Apply Changes**

The rule will look like Figure *Firewall rule to allow DHCP*.

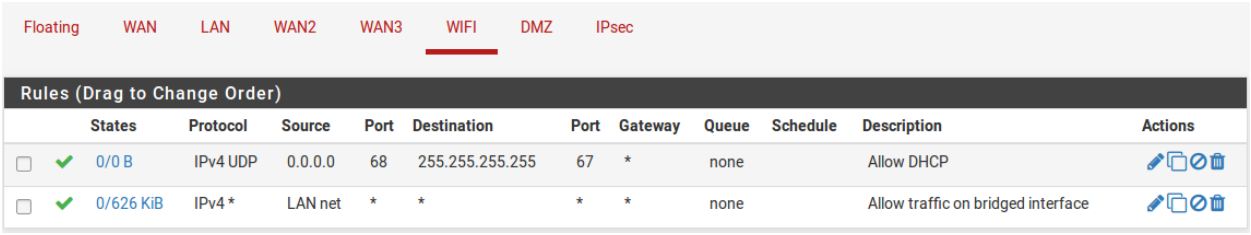


Fig. 2: Firewall rule to allow DHCP

After adding the rule, clients in the bridged segment will be able to successfully make requests to the DHCP daemon listening on the interface to which it is bridged.

DHCPv6 is a bit more complicated to allow since it communicates to and from both link-local and multicast IPv6 addresses. See Figure *Firewall Rule to Allow both DHCP and DHCPv6* for the list of required rules. These can be simplified with aliases into one or two rules containing the proper source network, destination network, and ports.

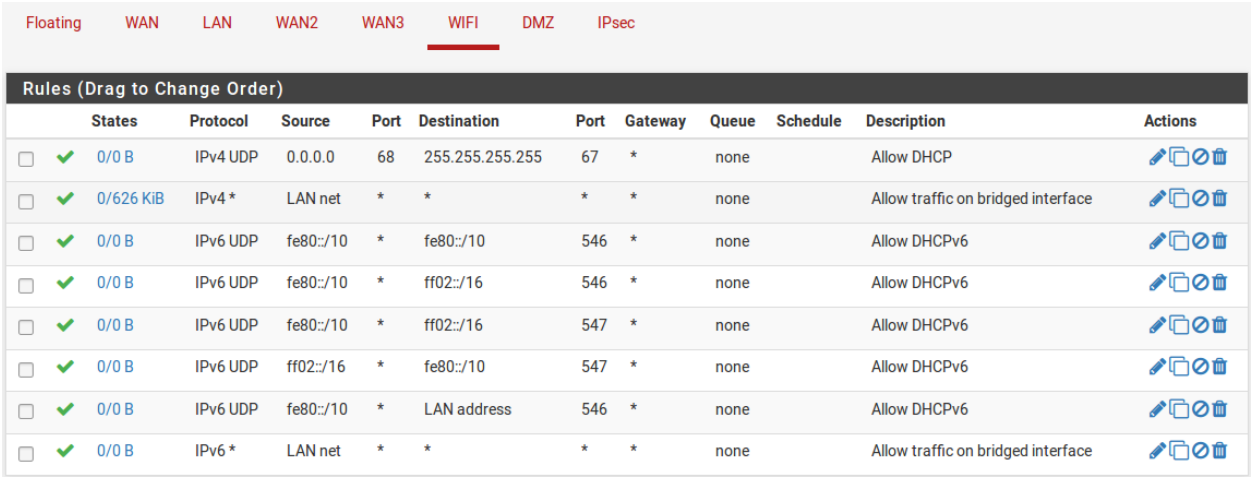


Fig. 3: Firewall Rule to Allow both DHCP and DHCPv6

16.6 Bridging interoperability

Bridged interfaces are different from normal interfaces, thus there are a few features that are incompatible with bridging and others where additional considerations are necessary to accommodate bridging. This section covers features that work differently with bridging than with non-bridged interfaces.

16.6.1 Captive portal

Captive portal (*Captive Portal*) is not compatible with transparent bridging because it requires an IP address on the interface being bridged, used to serve the portal contents, and that IP address must be the gateway for clients. This means that it is not possible, for example, to bridge LAN and WAN and hope to capture clients with the portal.

This can work when bridging multiple *local* interfaces to all route through pfSense® (e.g. LAN1, LAN2, LAN3, etc). It will work if the bridge interface is assigned, the bridge interface has an IP address, and that IP address is used as the gateway by clients on the bridge. See *Swapping Interface Assignments* for a procedure to place the IP address on an assigned bridge interface.

16.6.2 High Availability

High availability (*High Availability*) is not recommended with bridging. Some users have had mixed success with combining the two in the past but great care must be taken to handle layer 2 loops, which are unavoidable in a HA+bridge scenario. When two network segments are bridged, they are in effect merged into one larger network, as discussed earlier in this chapter. When HA is added into the mix, that means there will be two paths between the switches for each respective interface, creating a loop.

Managed switches can handle this with Spanning Tree Protocol (STP) but unmanaged switches have no defenses against looping. Left unchecked, a loop will bring a network to its knees and make it impossible to pass any traffic. STP may be configured on bridges to help, though there may still be unexpected results.

Consult switch documentation for information on STP configuration. Configure the switch to give preference to the port(s) on the primary node.

16.6.3 Multi-WAN

Transparent bridging by its nature is incompatible with multi-WAN in many of its uses. When using bridging between a WAN and LAN/OPT interface, commonly something other than pfSense will be the default gateway for the hosts on the bridged interface, and that router is the only device that can direct traffic from those hosts. This doesn't prevent multi-WAN from being used with other interfaces on the same firewall that are not bridged, it only impacts the hosts on bridged interfaces where they use something other than pfSense as their default gateway. If multiple internal interfaces are bridged together and pfSense is the default gateway for the hosts on the bridged interfaces, then multi-WAN can be used the same as with non-bridged interfaces.

16.6.4 Limiters

For limiters to function with bridging, the bridge itself must be assigned and the bridge interface must have the IP address and not a member interface.

16.6.5 LAN NAT and Transparent Proxies

For port forwards on LAN, or transparent proxies which use port forwards on LAN to capture traffic, to function in a bridge scenario, the situation is the same as Captive Portal: It will only function for LAN bridges and not WAN/LAN bridges, the IP address must be on the assigned bridge interface, and that IP address must be used as the gateway for local clients.

This means that a package such as Squid cannot work in a transparent firewall scenario where LAN is bridged to a WAN.

16.6.6 Mixing Bridged and NAT Segments

For hosts behind the NAT/routed segment, NAT must occur as traffic exits toward the bridged systems so that the return traffic will come back to the firewall.

For hosts on the bridged segment to reach hosts behind the NAT segment directly, a static route could be used on the bridged hosts or upstream gateway to send the "private" subnet traffic to the IP address of the firewall in the bridged network.

Normally each interface on the pfSense® firewall represents its own broadcast domain with a unique IP subnet. In some circumstances it is desirable or necessary to combine multiple interfaces onto a single broadcast domain, where two ports on the firewall will act as if they are on the same switch, except traffic between the interfaces can be controlled with firewall rules. Typically this is done so multiple interfaces will act as though they are on the same flat network using the same IP subnet and so that clients all share broadcast and multicast traffic.

Certain applications and devices rely on broadcasts to function, but these are found more commonly in home environments than corporate environments. For a practical discussion, see *Bridging and wireless*.

For services running on the firewall, bridging can be problematic. Features such as limiters, Captive Portal, and transparent proxies require special configuration and handling to work on bridged networks. Specifically, the bridge itself must be assigned and the only interface on the bridge with an IP address must be the assigned bridge. Also, in order for these functions to work, the IP address on the bridge must be the address used by clients as their gateway. These issues are discussed more in-depth in *Bridging interoperability*.

16.7 Types of Bridges

There are two distinct types of bridges: Internal bridges and Internal/external bridges. Internal bridges connect two local interfaces such as two LAN interfaces or a LAN interface and a wireless interface. Internal/external bridges connect a LAN to a WAN resulting in what is commonly called a "transparent firewall".

16.7.1 Internal Bridges

With an internal type bridge, ports on the firewall are linked such that they behave similar to switch ports, though with the ability to filter traffic on the ports or bridge and with much lower performance than a switch. The firewall itself is still visible to the local connected clients and acts as their gateway, and perhaps DNS and DHCP server. Clients on the bridged segments may not even know there is a firewall between them.

This type of configuration is commonly chosen by administrators to isolate and control a portion of the network, such as a wireless segment, or to make use of additional ports on the firewall in lieu of a proper switch where installing a switch would be impractical. Though it is not recommended, this type of bridge can also be used to join two remote networks over certain types of VPN connections.

See also:

pfSense Hangouts on Youtube to view the May 2015 Hangout on Wireless Access Points which included practical examples of internal type bridges.

16.7.2 Internal/External Bridges

An Internal/External type bridge, also known as a "transparent firewall", is used to insert a firewall between two segments without altering the other devices. Most commonly this is used to bridge a WAN to an internal network so that the WAN subnet may be used "inside" the firewall, or internally between local segments as an in-line filter. Another common use is for devices behind the firewall to obtain IP addresses via DHCP from an upstream server on the WAN.

In a transparent firewall configuration the firewall does not receive the traffic directly or act as a gateway, it merely inspects the traffic as it passes through the firewall.

Note: Devices on the internal side of this bridge **must** continue to use the upstream gateway as their own gateway. Do not set any IP address on the firewall as a gateway for devices on a transparent bridge.

NAT is not possible with this style of bridge because NAT requires the traffic to be addressed to the firewall's MAC address directly in order to take effect. Since the firewall is not the gateway, this does not happen. As such, rules to capture traffic such as those used by a transparent proxy do not function.

16.8 Bridging and Layer 2 Loops

When bridging, care must be taken to avoid layer 2 loops, or a switch configuration must be in place that handles loops. A layer 2 loop is when, either directly or indirectly, the switch has a connection back to itself. If a firewall running pfSense has interfaces bridged together, and two interfaces are plugged into the same switch on the same VLAN, a layer 2 loop has been created. Connecting two patch cables between two switches also does this.

Managed switches employ Spanning Tree Protocol (STP) to handle situations like this, because it is often desirable to have multiple links between switches, and the network shouldn't be exposed to complete meltdown by someone plugging one network port into another network port. STP is not enabled by default on all managed switches, and is almost never available with unmanaged switches. Without STP, the result of a layer 2 loop is frames on the network will circle endlessly and the network will completely cease to function until the loop is removed. Check the switch configuration to ensure the feature is enabled and properly configured.

pfSense enables STP on bridge interfaces to help with loops, but it can still lead to unexpected situations. For instance, one of the bridge ports would shut itself down to stop the loop, which could cause traffic to stop flowing unexpectedly or bypass the firewall entirely.

In a nutshell, bridging has the potential to completely melt down the network unless anyone that plugs devices into the switch is careful.

CHAPTER

SEVENTEEN

VIRTUAL LANS (VLANS)

17.1 Terminology

This section defines the terminology required to successfully deploy VLANs.

Trunking Trunking refers to a means of carrying multiple VLANs on the same physical switch port. The frames leaving a trunk port are marked with an 802.1Q tag in the header, enabling the connected device to differentiate between multiple VLANs. Trunk ports are used to connect multiple switches, and for connecting any devices that are capable of 802.1Q tagging and require access to multiple VLANs. This is commonly limited to the firewall or router providing connectivity between VLANs, in this case, the pfSense® firewall, as well as any connections to other switches containing multiple VLANs.

VLAN ID Each VLAN has an identifier number (ID) for distinguishing tagged traffic. This is a number between *1* and *4094*. The default VLAN on switches is VLAN *1*, and this VLAN should not be used when deploying VLAN trunking. This is discussed further in *VLANs and Security*. Aside from avoiding the use of VLAN 1, VLAN numbers may be chosen at will. Some designs start with VLAN 2 and increment by one until the required number of VLANs is reached. Another common design is to use the third octet in the subnet of the VLAN as the VLAN ID. For example, if 10.0.10.0/24, 10.0.20.0/24 and 10.0.30.0/24 are used, it is logical to use VLANs 10, 20, and 30 respectively. Choose a VLAN ID assignment scheme that makes sense for a given network design.

Parent interface The physical interface where a VLAN resides is known as its **Parent Interface**. For example, *igb0* or *em0*. When VLANs are configured on pfSense, each is assigned a virtual interface. The virtual interface name is crafted by combining the parent interface name plus the VLAN ID. For example, for VLAN 20 on *igb0*, the interface name is `igb0_vlan20`.

Note: The sole function of the parent interface is, ideally, to be the parent for the defined VLANs and not used directly. In some situations this will work, but can cause difficulties with switch configuration, and it requires use of the default VLAN on the trunk port, which is best to avoid as discussed further in *VLANs and Security*.

Access Port An access port refers to a switch port providing access to a single VLAN, where the frames are not tagged with an 802.1Q header. Normal client-type devices are connected to access ports, which will comprise the majority of switch ports. Devices on access ports do not need knowledge of VLANs or tagging. They see the network on their port the same as they would a switch without VLANs.

Double tagging (QinQ) *QinQ* refers to the double tagging of traffic, using both an outer and inner 802.1Q tag. This can be useful in large ISP environments, other very large networks, or networks that must carry multiple VLANs across a link that only supports a single VLAN tag. Triple tagging is also possible. pfSense supports QinQ, though it is not a very commonly used feature. These types

of environments generally need the kind of routing power that only a high end ASIC-based router can support, and QinQ adds a level of complexity that is unnecessary in most environments. For more information on configuring QinQ on pfSense, see *QinQ Configuration*.

Private VLAN (PVLAN) PVLAN, sometimes called **Port Isolation**, refers to capabilities of some switches to segment hosts within a single VLAN. Normally hosts within a single VLAN function the same as hosts on a single switch without VLANs configured. PVLAN provides a means of preventing hosts on a VLAN from talking to any other host on that VLAN, only permitting communication between that host and its default gateway. This isn't directly relevant to pfSense, but is a common question. Switch functionality such as this is the only way to prevent communication between hosts in the same subnet. Without a function like PVLAN, no network firewall can control traffic within a subnet because it never touches the default gateway.

17.2 VLANs and Security

VLANs are a great way to segment a network and isolate subnetworks, but there are security issues which need to be taken into account when designing and implementing a solution involving VLANs. VLANs are not inherently insecure, but misconfiguration can leave a network vulnerable. There have also been past security problems in switch vendor implementations of VLANs.

17.2.1 Segregating Trust Zones

Because of the possibility of misconfiguration, networks of considerably different trust levels should be on separate physical switches. For example, while the same switch could technically be used with VLANs for all internal networks as well as the network outside the firewalls, that should be avoided as a simple misconfiguration of the switch could lead to unfiltered Internet traffic entering the internal network. At a minimum, use two switches in such scenarios: One for outside the firewall and one inside the firewall. In many environments, DMZ segments are also treated separately, on a third switch in addition to the WAN and LAN switches. In others, the WAN side is on its own switch, while all the networks behind the firewall are on the same switches using VLANs. Which scenario is most appropriate for a given network depends on its specific circumstances, level of risk, and security concerns.

17.2.2 Using the default VLAN1

Because VLAN 1 is the default ("native") VLAN, it may be used in unexpected ways by the switch. It is similar to using a default-allow policy on firewall rules instead of default deny and selecting what is needed. Using a different VLAN is always better, and ensure that only the ports are selected that must be on that VLAN, to better limit access. Switches will send internal protocols such as STP (Spanning Tree Protocol), VTP (VLAN Trunking Protocol), and CDP (Cisco Discover Protocol) untagged over the native VLAN, where the switches use these protocols. It is generally best to keep that internal traffic isolated from data traffic.

If VLAN 1 must be used, take great care to assign every single port on every switch to a different VLAN except those that must be in VLAN 1, and do not create a management interface for the switch on VLAN 1. The native VLAN of the switch group should also be changed to a different, unused, VLAN. Some switches may not support any of these workarounds, and so it is typically easier to move data to a different VLAN instead of fussing with making VLAN 1 available. With VLAN ID 2 through 4094 to choose from, it is undoubtedly better to ignore VLAN 1 when designing a new VLAN scheme.

17.2.3 Using a trunk port's default VLAN

When VLAN tagged traffic is sent over a trunk on the native VLAN, tags in the packets that match the native VLAN may be stripped by the switch to preserve compatibility with older networks. Worse yet, packets that are double tagged with the native VLAN and a different VLAN will only have the native VLAN tag removed when trunking in this way and when processed later, that traffic can end up on a different VLAN. This is also called "VLAN hopping".

As mentioned in the previous section, any untagged traffic on a trunk port will be assumed to be the native VLAN, which could also overlap with an assigned VLAN interface. Depending on how the switch handles such traffic and how it is seen by pfSense®, using the interface directly could lead to two interfaces being on the same VLAN.

17.2.4 Limiting access to trunk ports

Because a trunk port can talk to any VLAN in a group of trunked switches, possibly even ones not present on the current switch depending on the switch configurations, it is important to physically secure trunk ports. Also make sure there are no ports configured for trunking that are left unplugged and enabled where someone could hook into one, accidentally or otherwise. Depending on the switch, it may support dynamic negotiation of trunking. Ensure this functionality is disabled or properly restricted.

17.2.5 Other Issues with Switches

Over the years there have been reports of rare cases where VLAN-based switches have leaked traffic across VLANs while under heavy loads, or if a MAC address of a PC on one VLAN is seen on another VLAN. These issues tend to be in older switches with outdated firmware, or extremely low-quality managed switches. These types of issues were largely resolved many years ago, when such security problems were common. No matter what switch from what brand is used for a network, research to see if it has undergone any kind of security testing, and ensure the latest firmware is loaded on the switch. While these issues are a problem with the switch, and not pfSense, they are part of a network's overall security.

Many of the items here are specific to particular makes and models of switches. Security considerations differ based on the switch being used on a network. Refer to its documentation for recommendations on VLAN security.

17.3 pfSense VLAN Configuration

This section covers how to configure VLANs in pfSense® software.

17.3.1 Console VLAN configuration

VLANs can be configured at the console using the **Assign Interfaces** function. The following example shows how to configure two VLANs, ID **10** and **20**, with **igb2** as the parent interface. The VLAN interfaces are assigned as **OPT1** and **OPT2**:

```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) pfSense Developer Shell
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Disable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

(continues on next page)

(continued from previous page)

```
Enter an option: 1

Valid interfaces are:

igb0   00:08:a2:09:95:b5   (up) Intel(R) PRO/1000 Network Connection, Version -
igb1   00:08:a2:09:95:b6   (up) Intel(R) PRO/1000 Network Connection, Version -
igb2   00:08:a2:09:95:b1 (down) Intel(R) PRO/1000 Network Connection, Version -
igb3   00:08:a2:09:95:b2 (down) Intel(R) PRO/1000 Network Connection, Version -
igb4   00:08:a2:09:95:b3 (down) Intel(R) PRO/1000 Network Connection, Version -
igb5   00:08:a2:09:95:b3 (down) Intel(R) PRO/1000 Network Connection, Version -

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? y

WARNING: all existing VLANs will be cleared if you proceed!

Do you want to proceed [y|n]? y

VLAN Capable interfaces:

igb0    00:08:a2:09:95:b5   (up)
igb1    00:08:a2:09:95:b6   (up)
igb2    00:08:a2:09:95:b1
igb3    00:08:a2:09:95:b2
igb4    00:08:a2:09:95:b3   (up)
igb5    00:08:a2:09:95:b3   (up)

Enter the parent interface name for the new VLAN (or nothing if finished): igb2
Enter the VLAN tag (1-4094): 10

VLAN Capable interfaces:

igb0    00:08:a2:09:95:b5   (up)
igb1    00:08:a2:09:95:b6   (up)
igb2    00:08:a2:09:95:b1
igb3    00:08:a2:09:95:b2
igb4    00:08:a2:09:95:b3   (up)
igb5    00:08:a2:09:95:b3   (up)

Enter the parent interface name for the new VLAN (or nothing if finished): igb2
Enter the VLAN tag (1-4094): 20

VLAN Capable interfaces:

igb0    00:08:a2:09:95:b5   (up)
igb1    00:08:a2:09:95:b6   (up)
igb2    00:08:a2:09:95:b1
igb3    00:08:a2:09:95:b2
igb4    00:08:a2:09:95:b3   (up)
igb5    00:08:a2:09:95:b3   (up)

Enter the parent interface name for the new VLAN (or nothing if finished): <enter>
```

(continues on next page)

(continued from previous page)

```
VLAN interfaces:

igb2_vlan10     VLAN tag 10, parent interface igb2
igb2_vlan20     VLAN tag 20, parent interface igb2

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(igb0 igb1 igb2 igb3 igb4 igb5 igb2_vlan10 igb2_vlan20 or a): igb1

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(igb0 igb2 igb3 igb4 igb5 igb2_vlan10 igb2_vlan20 a or nothing if finished): igb0

Enter the Optional 1 interface name or 'a' for auto-detection
(igb2 igb3 igb4 igb5 igb2_vlan10 igb2_vlan20 a or nothing if finished): igb2_vlan10

Enter the Optional 2 interface name or 'a' for auto-detection
(igb2 igb3 igb4 igb5 igb2_vlan20 a or nothing if finished): igb2_vlan20

Enter the Optional 3 interface name or 'a' for auto-detection
(igb2 igb3 igb4 igb5 a or nothing if finished):<enter>

The interfaces will be assigned as follows:

WAN  -> igb1
LAN  -> igb0
OPT1 -> igb2_vlan10
OPT2 -> igb2_vlan20

Do you want to proceed [y|n]? y

Writing configuration...done.
One moment while the settings are reloading... done!
```

After a few seconds, the firewall settings will reload and the console menu will reload.

17.3.2 Web interface VLAN configuration

In the system used for this example, WAN and LAN are assigned as *igb1* and *igb0* respectively. There is also an *igb2* interface that will be used as the VLAN parent interface.

To configure VLANs in the pfSense web interface:

- Navigate to **Interfaces > Assignments** to view the interface list.
- Click the **VLANs** tab.
- Click  **Add** to add a new VLAN
- Configure the VLAN as shown in Figure *Edit VLAN*.

  Parent Interface The physical interface upon which this VLAN tag will be used. In this case, *igb2*

VLAN tag The VLAN ID number, in this case, `10`

VLAN Priority Leave at the default value, blank

Description Some text to identify the purpose of the VLAN, such as `DMZ`

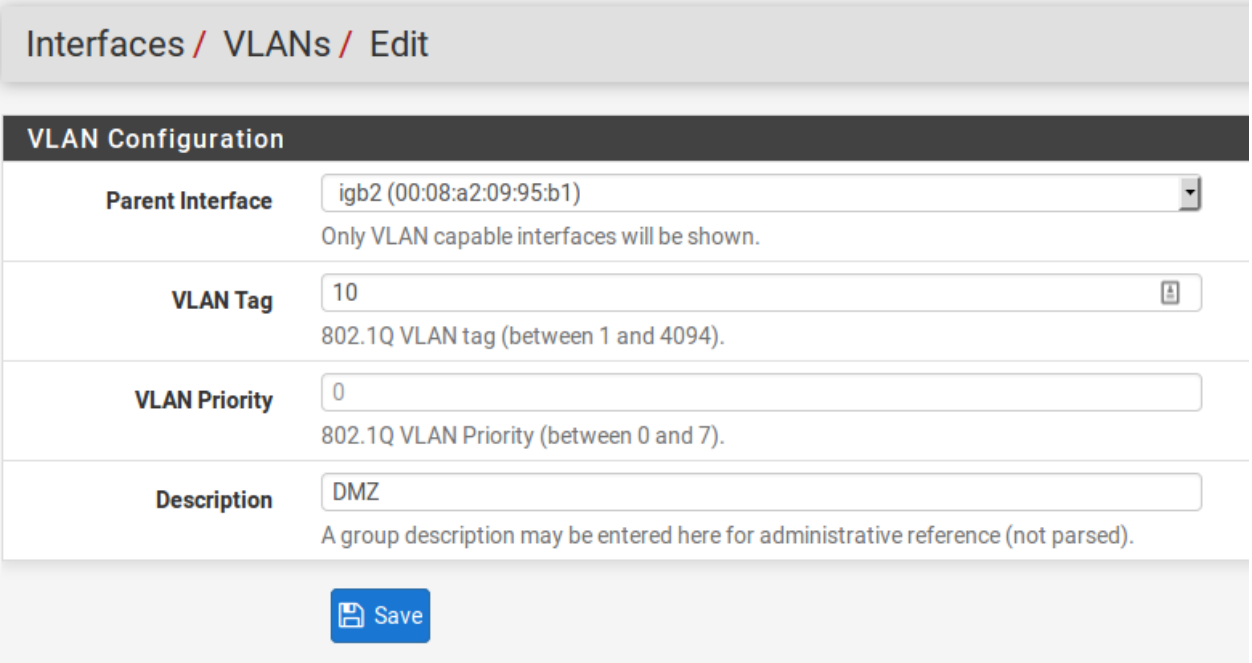


Fig. 1: Edit VLAN

- Click **Save** to return to the VLAN list, which now includes the newly added VLAN 10.
- Repeat the process to add additional VLANs, such as VLAN 20. These can be seen in Figure *VLAN list*

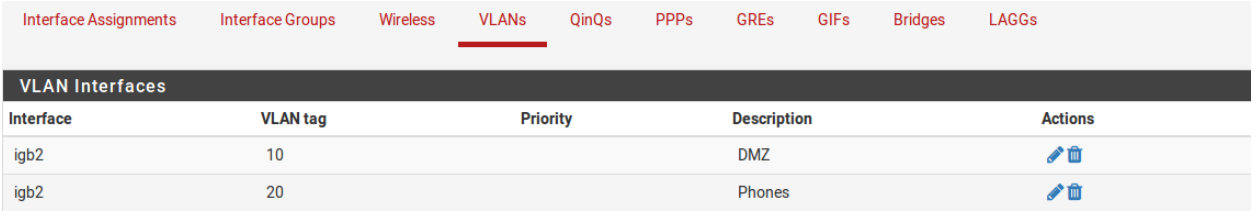


Fig. 2: VLAN list

To assign the VLANs to interfaces:

- Navigate to **Interfaces > Assignments**
- Click the **Interface Assignments** tab
- Select the VLAN to add from the **Available Network Ports** list, such as *VLAN 10 on igb2 (DMZ)*
- Click **Add** to assign the network port
- Repeat the last two steps to assign *VLAN 20 on igb2 (Phones)*

When finished, the interfaces will look like Figure *Interfaces list with VLANs*

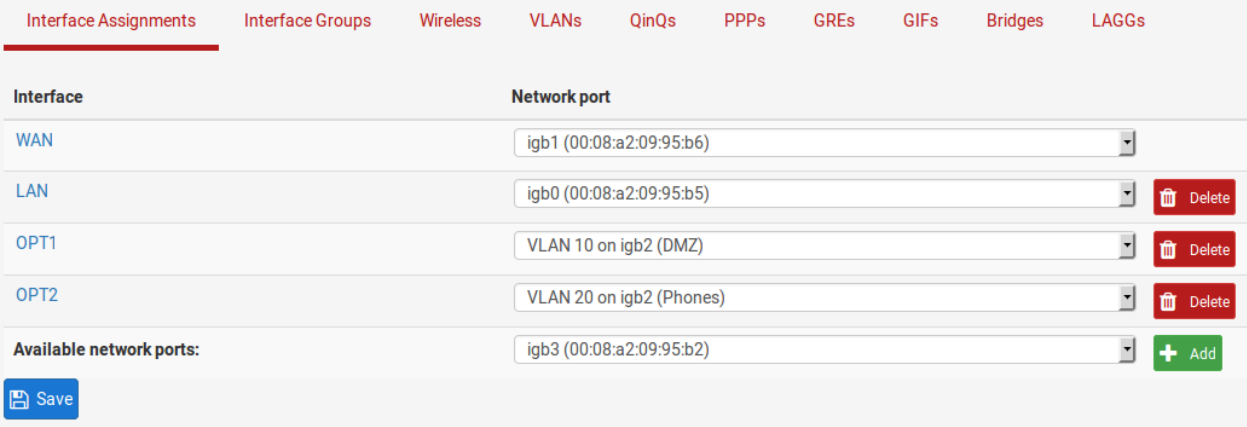


Fig. 3: Interfaces list with VLANs

The VLAN-based OPT interfaces behave as any other OPT interfaces do, which means they must be enabled, configured, have firewall rules added, and services like the DHCP Server will need to be configured if needed. See *Interface Configuration Basics* for more information on configuring optional interfaces.

See also:

- *Configuring Switches with VLANs*
- *QinQ Configuration*

VLANs enable a switch to carry multiple discrete broadcast domains, allowing a single switch to function as if it were multiple switches. VLANs are commonly used for network segmentation in the same way that multiple switches can be used: To place hosts on a specific segment, isolated from other segments. Where trunking is employed between switches, devices on the same segment need not reside on the same switch. Devices that support trunking can also communicate on multiple VLANs through a single physical port.

This chapter covers VLAN concepts, terminology and configuration in pfSense® software.

17.4 Requirements

There are two requirements, both of which must be met to deploy VLANs.

1. **802.1Q VLAN capable switch**

   Every decent managed switch manufactured in the last 15 years supports 802.1Q VLAN trunking.

   > **Warning:** VLANs **cannot** be used with an unmanaged switch.

2. **Network adapter capable of VLAN tagging**

   A NIC that supports hardware VLAN tagging or has long frame support is required. Each VLAN frame has a 4 byte 802.1Q tag added in the header, so the frame size can be up to 1522 bytes. A NIC supporting hardware VLAN tagging or long frames is required because other adapters will not function with frames larger than the normal 1518 byte maximum with 1500 MTU Ethernet. This will cause large frames to be dropped, which causes performance problems and connection stalling.

   Note: If an adapter is listed as having long frame support does not guarantee the specific implementation of

that NIC chipset properly supports long frames. Realtek *rl(4)* NICs are the biggest offenders. Many will work fine, but some do not properly support long frames, and some will not accept 802.1Q tagged frames at all. If problems are encountered using one of the NICs listed under long frame support, we recommend trying an interface with VLAN hardware tagging support instead. We are not aware of any similar problems with NICs listed under VLAN hardware support.

Ethernet interfaces with VLAN hardware support:

ae(4), *age(4)*, *alc(4)*, *ale(4)*, *bce(4)*, *bge(4)*, *bxe(4)*, *cxgb(4)*, *cxgbe(4)*, *em(4)*, *igb(4)*, *ixgb(4)*, *ixgbe(4)*, *jme(4)*, *msk(4)*, *mxge(4)*, *nxge(4)*, *nge(4)*, *re(4)*, *sge(4)*, *stge(4)*, *ti(4)*, *txp(4)*, *vge(4)*.

Ethernet interfaces with long frame support :

axe(4), *bfe(4)*, *cas(4)*, *dc(4)*, *et(4)*, *fwe(4)*, *fxp(4)*, *gem(4)*, *hme(4)*, *le(4)*, *nfe(4)*, *nve(4)*, *rl(4)*, *sf(4)*, *sis(4)*, *sk(4)*, *ste(4)*, *tl(4)*, *tx(4)*, *vr(4)*, *vte(4)*, *xl(4)*.

CHAPTER

EIGHTEEN

MULTIPLE WAN CONNECTIONS

18.1 Multi-WAN Terminology and Concepts

This section covers the terminology and concepts necessary to understand to deploy multi-WAN with pfSense® software.

18.1.1 WAN-type Interface

A WAN-type interface is an interface through which the Internet can be reached, directly or indirectly. The firewall treats any interface with a gateway selected on its **Interfaces** menu page as a WAN. For example, with a static IP address WAN, **Interfaces > WAN** has a gateway selected, such as **WAN_GW**. If this gateway selection is not present, then the interface will be treated as a local interface instead. Dynamic IP address interfaces such as DHCP and PPPoE receive a dynamic gateway automatically and are always treated as WANs.

The presence of a gateway on the interface configuration changes the firewall behavior on such interfaces in several ways:

- Firewall rules on these interfaces have `reply-to` added which returns connections coming in through that WAN back out via the same WAN where possible
- These interfaces are used as exit interfaces for automatic and hybrid outbound NAT
- These interfaces are treated as WANs by the traffic shaper wizard

Warning: Do not select a gateway on the **Interfaces** menu entry for local interfaces such as LAN.

Local and other interfaces may have a gateway defined under **System > Routing**, so long as that gateway is not chosen under their interface configuration, for example on **Interfaces > LAN**.

18.1.2 Policy routing

Policy routing refers to a means of routing traffic by more than the destination IP address of the traffic, as is done with the routing table in most operating systems and routers. This is accomplished by the use of a policy of some sort, usually firewall rules or an access control list. In pfSense, the **Gateway** field available when editing or adding firewall rules enables the use of policy routing. The **Gateway** field contains all gateways defined on the firewall under **System > Routing**, plus any gateway groups.

Policy routing provides a powerful means of directing traffic to the appropriate WAN interface or other gateway, since it allows matching anything a firewall rule can match. Specific hosts, subnets, protocols and more can be used to direct traffic.

Note: Remember that all firewall rules, including policy routing rules, are processed in top down order, and the first match wins.

18.1.3 Gateway Groups

Gateway groups define how a chosen set of gateways provide failover and/or load balancing functionality. They are configured under **System > Routing**, on the **Gateway Groups** tab.

See *Gateway Groups* for more.

18.1.4 Failover

Failover refers to the ability to switch from one or more WANs to an alternate WAN if the preferred connection fails. This is useful for situations where traffic should utilize one specific WAN connection unless it is unavailable.

See also:

To fail from one *firewall* to another, rather than from one *WAN* to another, see *High Availability*.

18.1.5 Load Balancing

The **Load Balancing** functionality in pfSense software distributes connections over multiple WAN connections in a round-robin fashion. This feature operates on a **per-connection** basis. If a gateway that is part of a load balancing group fails, the interface is marked as down and removed from all groups until it recovers, thus a load balanced configuration effectively also includes failover functionality.

18.1.6 Monitor IP Addresses

When configuring failover or load balancing, each gateway is associated with a monitor IP address (*Gateway Settings*). In a typical configuration, the firewall will ping this IP address and if it stops responding, the gateway is marked as down. Options on the gateway group can select different failure triggers besides packet loss. The other triggers are high latency, a combination of either packet loss or high latency, or when the circuit is down.

What constitutes failure?

The topic is a little more complex than "if pings to the monitor IP address fail, the gateway is marked as down." The actual criteria for a failure depend on the options chosen when creating the gateway group and the individual settings on a gateway.

The settings for each gateway that control when it is considered up and down are all discussed in *Advanced Gateway Settings*. The thresholds for packet loss, latency, down time, and even the probing interval of the gateway are all individually configurable.

18.1.7 State Killing/Forced Switch

When a gateway has failed, the firewall can optionally flush all states to force clients to reconnect, and in doing so they will use a gateway that is online instead of a gateway that is down. This currently only works one-way, meaning that it can move connections off of a failing gateway, but it cannot force them back if the original gateway comes back online.

This is an optional behavior, but it is **not** enabled by default since it is disruptive. For information on changing this setting, see *Gateway Monitoring*.

18.1.8 Default Gateway Switching

Traffic exiting the firewall itself will use the default gateway unless a static route sends the packet along a different path. If the default gateway is on a WAN that is down, daemons on the firewall will be unable to make outbound connections, depending on the capabilities of the daemon and its configuration.

The default gateway for the firewall can be set to a gateway group or set to an automatic mode, which will switch the default to the next available gateway if the normal default gateway fails, and then switched back when that WAN recovers. See *Managing the Default Gateway* for details.

18.2 Policy Routing, Load Balancing and Failover Strategies

This section provides guidance on common Multi-WAN goals and how they are achieved with pfSense® software.

18.2.1 Bandwidth Aggregation

One of the primary desires with multi-WAN is bandwidth aggregation. Load balancing can help accomplish this goal. There is, however, one caveat: If the firewall has two 50 Mbps WAN circuits, it cannot get 100 Mbps of throughput with a *single* client connection. Each individual connection must be tied to only one specific WAN. This is true of any multi-WAN solution other than MLPPP. The bandwidth of two different Internet connections cannot be aggregated into a single large "pipe" without involvement from the ISP. With load balancing, since individual connections are balanced in a round-robin fashion, 100 Mbps of throughput can only be achieved using two 50 Mbps circuits when multiple connections are involved. Applications that utilize multiple connections, such as download accelerators, will be able to achieve the combined throughput capacity of the two or more connections.

Note: Multi-Link PPPoE (MLPPP) is the only WAN type which can achieve full aggregate bandwidth of all circuits in a bundle, but requires special support from the ISP. For more on MLPPP, see *Multi-Link PPPoE (MLPPP)*

In networks with numerous internal machines accessing the Internet, load balancing will reach speeds near the aggregate throughput by balancing the many internal connections out all of the WAN interfaces.

18.2.2 Segregation of Priority Services

Consider a site which has a reliable, high quality Internet connection that offers low bandwidth, or high costs for excessive transfers, and another connection that is fast but of lesser quality (higher latency, more jitter, or less reliable). In these situations, services can be segregated between the two Internet connections by their priority. High priority services may include VoIP, traffic destined to a specific network such as an outsourced application provider, or specific protocols used by critical applications, amongst other options. Low priority traffic commonly includes any permitted traffic that doesn't match the list of high priority traffic. Policy routing rules can be setup to direct the high priority traffic out the high quality Internet connection, and the lower priority traffic out the lesser quality connection.

Another example of a similar scenario is getting a dedicated Internet connection for quality critical services such as VoIP, and only using that connection for those services.

18.2.3 Failover Only

There are scenarios where only using failover is the best practice. For example, users who have a secondary backup Internet connection with a low bandwidth cap such as a 4G/LTE modem, and only want to use that connection if their primary connection fails, Gateway groups configured for failover can achieve this goal.

Another usage for failover is to ensure a certain protocol or destination always uses only one WAN unless it goes down.

18.2.4 Unequal Cost Load Balancing

pfSense software can achieve unequal cost load balancing by setting appropriate weights on the gateways as discussed in *Advanced Gateway Settings*. By setting a weight on a gateway, it will be used more often in a gateway group. Weights can be set from *1* to *30*, allowing

Table 1: Unequal Cost Load Balancing

WAN_GW weight	WAN2_GW weight	WAN load	WAN2 load
3	2	60%	40%
2	1	67%	33%
3	1	75%	25%
4	1	80%	20%
5	1	83%	17%
30	1	97%	3%

Note that this distribution is strictly balancing the number of *connections*, it does not take interface throughput into account. This means bandwidth usage will not necessary be distributed equally, though in most environments it works out to be roughly distributed as configured over time. This also means if an interface is loaded to its capacity with a single high throughput connection, additional connections will still be directed to that interface.

18.3 Multi-WAN Caveats and Considerations

This section contains the caveats and considerations specific to multi-WAN in pfSense® software.

18.3.1 Multiple WANs sharing a single gateway IP

Due to the way pf handles multi-WAN connections, traffic can only be directed using the gateway IP address of a circuit, which is fine for most scenarios. If the firewall has multiple connections on the same ISP using the same subnet and gateway IP address, as is common when using multiple cable modems, an intermediate NAT device must be used on all but one of them so that the firewall sees each WAN gateway as a unique IP address.

When using the NAT device, it can be configured to forward all traffic back to the firewall which can help with using that WAN for other services. However, some protocols, such as VoIP, will have problem if they use a WAN with NAT in such a configuration.

If at all possible, contact the ISP and have them configure the WAN circuits such that they are in different subnets with different gateways.

One exception to this is a PPP type WAN such as PPPoE. PPP type WANs are capable of having the same gateway on multiple interfaces, but each gateway entry must be configured to use a different monitor IP address (See *Gateway Settings*).

18.3.2 Multiple PPPoE WANs

When multiple PPPoE lines from the same ISP are present and the ISP supports Multi-Link PPPoE (MLPPP), it may be possible to bond the lines into a single aggregate link. This bonded link has total bandwidth of all lines together in a single WAN as seen by the firewall. Configuration of MLPPP is covered in *Multi-Link PPPoE (MLPPP)*.

18.3.3 Local Services and Multi-WAN

There are additional considerations with local services and multi-WAN, since any traffic initiated from the firewall itself will not be affected by policy routing configured on internal interface rules. Traffic from the firewall itself always follows the routing table. Hence static routes are required under some circumstances when using additional WAN interfaces, otherwise only the WAN interface with the default gateway would be used.

The firewall can be configured to change the default gateway if the preferred default fails. See *Managing the Default Gateway* for details.

In the case of traffic initiated on the Internet destined for any WAN interface, pfSense automatically uses the `reply-to` directive in all WAN-type interface rules, which ensures the reply traffic is routed back out the correct WAN interface.

DNS Resolver

The default settings for the DNS Resolver require using failover for the default gateway to work properly with Multi-WAN. See *Managing the Default Gateway* for details. As an alternative to using default gateway switching, a few changes can be made to make the DNS Resolver more accommodating to Multi-WAN, including enabling forwarding mode. The details are described later in this chapter.

DNS Forwarder

The DNS servers used by the DNS forwarder must have gateways defined if they use an non-default WAN interface, as described later in this chapter. There are no other caveats to DNS forwarder in multi-WAN environments.

Dynamic DNS

Dynamic DNS entries can be set using a gateway group for their interface. This will move a Dynamic DNS entry between WANs in failover mode, allowing a public hostname to shift from one WAN to another in case of failure.

IPsec

IPsec is fully compatible with multi-WAN. A static route is automatically added for the remote tunnel peer address pointing to the specified WAN gateway to ensure the firewall sends traffic out the correct path when it initiates a connection. For mobile connections, the client always initiates the connection, and the reply traffic is correctly routed by the state table.

An IPsec tunnel may also be set using a gateway group as its interface for failover. This is discussed further in *Multi-WAN Environments*.

OpenVPN

OpenVPN multi-WAN capabilities are described in *OpenVPN and Multi-WAN*. Like IPsec, it can use any WAN or a gateway group.

CARP and multi-WAN

CARP is multi-WAN capable so long as all WAN interfaces use static IP addresses and there are at least three public IP addresses available per WAN. This is covered in *High Availability Configuration Example with Multi-WAN*.

18.3.4 IPv6 and Multi-WAN

IPv6 is also capable of performing in a multi-WAN capacity, but will usually require Network Prefix Translation (NPt) on one or more WANs. This is covered in more detail in *Configuring Multi-WAN for IPv6*.

18.4 Summary of Multi-WAN Requirements

Before covering the bulk of multi-WAN specifics, here is a short summary of the requirements to make a fully implemented multi-WAN setup:

- Create a gateway group under **System > Routing** on the **Groups** tab
- Set a failover gateway group for the default gateway as described in *Managing the Default Gateway* (Technically optional but a best practice)
- Configure the DNS Resolver or Forwarder for Multi-WAN, starting by:
  - Use a failover gateway group for the default gaetway (DNS Resolver in default resolver mode)
  - Set at least one unique DNS server for each WAN gateway under **System > General Setup** with a gateway set (DNS Resolver in forwarding mode or DNS Forwarder)

- Use the gateway group on LAN firewall rules

18.5 Load Balancing and Failover with Gateway Groups

A Gateway Group is necessary to setup a Load Balancing or Failover configuration. The group itself does not cause any action to be taken, but when the group is used later, such as in policy routing firewall rules, it defines how the items utilizing the group will behave.

The same gateway may be included in multiple groups so that several different scenarios can be configured at the same time. For example, some traffic can be load balanced, and other traffic can use failover, and the same WAN can be used in both capacities by using different gateway groups.

A common example setup for a two WAN firewall contains three groups:

- **LoadBalance**: Gateways for WAN1 and WAN2 both on Tier 1
- **PreferWAN1**: Gateway for WAN1 on Tier 1, and WAN2 on Tier 2
- **PreferWAN2**: Gateway for WAN1 on Tier 2, and WAN2 on Tier 1

No matter which strategy is chosen, the best practice is to have at least one failover group and to set that failover group to be used as the default gateway on the firewall. This ensures that the firewall always has a viable default gateway, and using a gateway group ensures that the correct gateways are used for this function and in the intended order. See *Managing the Default Gateway* for details.

18.5.1 Configuring a Gateway Group for Load Balancing or Failover

To create a gateway group for Load Balancing or Failover:

- Navigate to **System > Routing**, **Gateway Groups** tab
- Click  **Add** to create a new gateway group
- Fill in the options on the page as described in *Gateway Group Options*
- Click **Save**

Load Balancing

Any two gateways on the same tier are load balanced. For example, if *Gateway A*, *Gateway B*, and *Gateway C* are all Tier 1, connections would be balanced between them. Gateways that are load balanced will automatically failover between each other. When a gateway fails it is removed from the group, so in this case if any one of A, B, or C went down, the firewall would load balance between the remaining online gateways.

Weighted Balancing

If two WANs need to be balanced in a weighted fashion due to differing amounts of bandwidth between them, that can be accommodated by adjusting the **Weight** parameter on the gateway as described in *Unequal Cost Load Balancing* and *Advanced Gateway Settings*.

Failover

Gateways on a **lower** number tier are preferred by the firewall, and if they are down then gateways of a higher numbered tier are used. For example, if *Gateway A* is on Tier 1, *Gateway B* is on Tier 2, and *Gateway C* is on Tier 3, then *Gateway A* would be used first. If *Gateway A* goes down, then *Gateway B* would be used. If both *Gateway A* and *Gateway B* are down, then *Gateway C* would be used.

Complex/Combined Scenarios

By extending the concepts above for Load Balancing and Failover, complicated scenarios are possible that combine both load balancing and failover. For example, if *Gateway A* is on Tier 1, and *Gateway B* and *Gateway C* are on Tier 2, then *Gateway D* on Tier 3, the following behavior occurs: *Gateway A* is preferred on its own. If *Gateway A* is down, then traffic would be load balanced between *Gateway B* and *Gateway C*. Should either *Gateway B* or *Gateway C* go down, the remaining online gateway in that tier would still be used. If *Gateway A*, *Gateway B*, and *Gateway C* are all down, traffic would fail over to *Gateway D*.

Any other combination of the above can be used, so long as it can be arranged within the limit of **5** tiers.

18.5.2 Problems with Load Balancing

Some websites store session information including the client IP address, and if a subsequent connection to that site is routed out a different WAN interface using a different public IP address, the website will not function properly. This is becoming more common with banks and other security-minded sites. One method of working around this issue is to create a failover group and direct traffic destined to these sites to the failover group rather than a load balancing group. Alternately, perform failover for all HTTPS traffic.

The sticky connections feature of pf is intended to resolve this problem, but it has historically been problematic. It is safe to use, and should alleviate this, but there is also a downside to using the sticky option. When using sticky connections, an association is held between the *client IP address* and a given *gateway*, it is not based off of the destination. When the sticky connections option is enabled, any given client would not load balance its connections between multiple WANs, but it would be associated with whichever gateway it happened to use for its first connection. Once all of the client states have expired, the client may exit a different WAN for its next connection, resulting in a new gateway pairing. As such, it works best in environments with many clients where one client utilizing a single WAN does not have a large impact.

18.6 Interface and DNS Configuration

The first two items to configure for Multi-WAN are Interfaces and DNS.

18.6.1 Interface Configuration

Setup the primary WAN as previously described in *Setup Wizard*. Then for the additional WAN interfaces, perform the following tasks:

- Assign the interfaces if they do not yet exist
- Visit the **Interfaces** menu entry for each additional WAN (e.g. **Interfaces > OPT1**)
- Enable the interface
- Enter a suitable name, such as `WAN2`
- Select the desired type of IP address configuration depending on the Internet connection type.
- Enter the remaining details for the type of WAN. For example, on static IP connections, fill in the IP address, subnet mask, and add or select a gateway.

18.6.2 DNS Server Configuration

If the DNS Resolver will be used in **forwarding** mode or if the DNS Forwarder is in use, the firewall must be configured with DNS servers from each WAN connection to ensure it is always able to resolve DNS. This is especially important if the internal network uses the firewall for DNS resolution.

If the firewall configuration only includes DNS servers from a single WAN, an outage of that WAN connection will result in a complete Internet outage regardless of policy routing configuration since DNS will no longer function.

18.6.3 DNS Resolver Configuration

The DNS Resolver can work with Multi-WAN but the exact configuration depends on the desired behavior and current settings.

If the DNS Resolver must work in its default resolver mode, such as for environments which require DNSSEC, then forwarding mode cannot be enabled. This can still function with Multi-WAN but requires using failover for the default gateway. See *Managing the Default Gateway*.

If the DNS Resolver can use forwarding mode, then the following procedure may be performed instead:

- Set at least one DNS server per WAN under **System > General Setup**, as described in the next section
- Check **Enable Forwarding Mode** under **Services > DNS Resolver**
- Uncheck **Enable DNSSEC Support**

DNS Servers and Static Routes

When using the DNS Forwarder or the DNS Resolver in forwarding mode, the firewall uses its routing table to reach the configured DNS servers. This means without any static routes configured, it will only use the primary WAN connection to reach DNS servers. Gateways must be selected for each DNS server defined on the firewall to use the correct WAN interface to reach that DNS server. DNS servers that come from dynamic gateways are automatically routed back out the proper path. At least one gateway from each WAN should be selected where possible.

To configure the DNS server gateways:

- Navigate to **System > General Setup**
- Define at least one *unique* DNS server for each WAN
- For each DNS server, select an appropriate gateway so it uses a specific WAN interface

Note: The same DNS server cannot be entered more than once. Each entry must be unique.

Selecting gateways for DNS servers is required for several reasons. One, most ISPs prohibit recursive queries from hosts outside their network, hence the firewall must use the correct WAN interface when accessing DNS servers for a specific ISP. Secondly, if the primary WAN fails and the firewall does not have a gateway chosen for one of the other DNS servers, the firewall will lose all DNS resolution ability from the firewall itself. Access to DNS is lost in that situation because all DNS servers will be unreachable when the default gateway is unreachable. If the firewall is used as a DNS server for the local network, this will result in a complete failure of DNS.

When using the DNS Resolver with forwarding mode disabled, the `unbound` daemon speaks directly to the root DNS servers and other authoritative DNS servers, which makes using such static routes and gateway assignments impossible. In that case, configure failover for the default gateway (*Managing the Default Gateway*) so that the DNS Resolver can maintain outbound connectivity.

18.6.4 Scaling to Large Numbers of WAN Interfaces

There are numerous users of pfSense® software deploying 6-12 Internet connections on a single installation. One user has 10 DSL lines because in his country it is significantly cheaper to get ten 256 Kb connections than it is one 2.5 Mb connection. That customer load balances a large number of internal machines out 10 different connections. For more information on this scale of deployment, see *Multi-WAN on a Stick* later in this chapter.

18.7 Multi-WAN and NAT

The default NAT rules generated by pfSense® software will translate any traffic leaving a WAN-type interface to the IP address of that interface. In a default two interface LAN and WAN configuration, pfSense will NAT all traffic from the LAN subnet leaving the WAN interface to the WAN IP address. Adding more WAN-type interfaces extends this to NAT any traffic leaving a WAN-type interface to that interface IP address. This is all handled automatically unless Manual Outbound NAT is enabled.

Warning: NAT does not influence the path taken by connections, only how addresses on packets traversing an interface are translated by the firewall.

Policy routing firewall rules direct connections to specific WAN interfaces, and the Outbound and 1:1 NAT rules specify how the addresses on packets for those connections will be translated by the firewall as it leaves that WAN.

18.7.1 Multi-WAN and Manual Outbound NAT

If **Manual Outbound NAT** must be used with multi-WAN, ensure manual outbound NAT rules are present on all WAN-type interfaces.

18.7.2 Multi-WAN and Port Forwarding

Each port forward applies to a single WAN interface. A given port can be opened on multiple WAN interfaces by using multiple port forward entries, one per WAN interface. The easiest way to accomplish this is:

- Add a port forward on the first WAN connection as usual
- Click  to the right of that entry to add another port forward based on the selected one
- Change the **Interface** to the desired WAN
- Click **Save**

The `reply-to` keyword in pf, used on WAN-type interface rules, ensures that when traffic comes in over a specific WAN interface, the return traffic will go back out the way it came into the firewall. So port forwards can be actively used on all WAN interfaces at any time, regardless of any failover configuration that may be present. This is especially useful for mail servers, as an address on a secondary WAN can be used as a backup MX, allowing the site to receive mail even when the primary line is down. This behavior is configurable, for information on this setting, see *Disable Reply-To*.

18.7.3 Multi-WAN and 1:1 NAT

1:1 NAT entries are specific to a single WAN interface and, like outbound NAT, they only control what happens to addresses on packets as they pass through an interface. Internal systems can be configured with a 1:1 NAT entry on each WAN interface, or a 1:1 entry on one or more WAN interfaces and use the default outbound NAT on others. Where 1:1 entries are configured, they always override any other Outbound NAT configuration for that specific interface.

If a local device must always use a 1:1 NAT entry on a specific WAN, then traffic from that device must be forced to use that specific WAN gateway with policy routing firewall rules.

18.8 Policy Routing Configuration

At this point, the firewall is prepared for Multi-WAN but it will not yet be used. Traffic will not properly fail over or be load balanced without policy routing firewall rules in place.

Note: An exception to this is when using a gateway group or automatic failover for the default gateway (*Managing the Default Gateway*). In this case failover could still function without policy routing, but not load balancing.

18.8.1 Configuring Firewall Rules for Policy Routing

Setting a **Gateway** on a firewall rule will cause traffic matching the rule to use the chosen gateway or group, following the configured behavior of the group.

The easiest way to configure a firewall for policy routing is to edit the existing default pass rule for the LAN and select the gateway group there. With that set, any traffic matching the default pass rule on the LAN will use the chosen gateway or group.

To make that edit:

- Navigate to **Firewall > Rules**, **LAN** tab

- Click  on the row with the default pass rule
- Click  **Display Advanced** under **Extra Options**
- Select the desired gateway group from the **Gateway** drop-down list
- Click **Save**
- Click **Apply Changes**

Only the most basic of deployments will be satisfied with that configuration, most configurations are more complex. Continue reading for more factors that can require additional configuration.

18.8.2 Bypassing Policy Routing

If there are other local interfaces, VPNs, MPLS interfaces, or traffic that must otherwise follow the system routing table, then that traffic must be configured to bypass policy routing. This is simple to do by making a rule to match the traffic in question and then placing that rule **above** any rules that have a gateway configured, because the first rule to match is the one that is used.

This can be generalized by making an alias for any *RFC1918* traffic which would cover all private networks, and then using that in a rule. The alias contains `192.168.0.0/16`, `172.16.0.0/12`, and `10.0.0.0/8`.

In Figure *Bypass Policy Routing Example Rules*, local and VPN traffic bypasses policy routing, HTTPS traffic prefers WAN2, and all other traffic is load balanced:

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0/0 B	*	*	*	LAN Address	443 80 22	*	*		Anti-Lockout Rule	⚙
☐ ✓	0/0 B	IPv4 *	LAN net	*	RFC1918	*	*	none		Bypass policy routing for local/VPN traffic	⚓✎⧉⊘🗑
☐ ✓ ⚙	0/0 B	IPv4 TCP	LAN net	*	*	443 (HTTPS)	PreferWAN2	none		Force HTTPS out WAN2, fail to WAN1	⚓✎⧉⊘🗑
☐ ✓ ⚙	0/10 KiB	IPv4 *	LAN net	*	*	*	LoadBalance	none		Load Balance LAN Traffic	⚓✎⧉⊘🗑

Fig. 1: Bypass Policy Routing Example Rules

18.8.3 Mixing Failover and Load Balancing

As shown in Figure *Bypass Policy Routing Example Rules*, failover and load balancing can be used at the same time by carefully ordering the rules on an interface. Rules are processed from the top down and the first match wins. By placing more specific rules near the top of the list, and the general "match all" style rules at the bottom, any number of different combinations are possible with rules using different gateways or groups.

18.8.4 Enforcing Gateway Use

There are situations where traffic should only ever use one gateway and never load balance or failover. In this example, a device must only exit via a specific WAN and lose all connectivity when that WAN fails.

First, set the **Gateway** on a firewall rule matching traffic from this device to a specific WAN Gateway. If that gateway is down, the rule will act as if the gateway was not set at all, so it needs to be taken a couple steps further.

Add a rule immediately below the rule matching the traffic, but set to reject or block instead. This rule must not have a gateway set.

Next, configure the firewall to omit rules for gateways that are down (*Gateway Monitoring*):

- Navigate to **System > Advanced** on the **Miscellaneous** tab
- Check **Do not create rules when gateway is down**
- Click **Save**

With that option enabled, the first rule will be omitted entirely, falling through to the next matching rule. This way, when the first rule is omitted automatically, traffic will be stopped by the block rule.

18.9 Verifying Functionality

Once multi-WAN has been configured, best practice is then to test its functionality to verify it functions as expected. The following sections describe how to test each portion of a multi-WAN configuration.

18.9.1 Testing Failover

Testing Multi-WAN in a controlled manner immediately after configuration is a key step in the process. Do not make the mistake of waiting until an Internet connection fails naturally for the first test, only to discover problems when they are much more difficult and stressful to fix.

First, navigate to **Status > Gateways** and ensure all WAN gateways are show as **Online** under **Status**, as well as on the **Gateway Groups** tab. If they do not, verify that a proper monitor IP address is used as discussed in *Gateway Settings*.

Creating a WAN Failure

There are a number of ways to simulate a WAN failure depending on the type of Internet connection being used. For any type, first try unplugging the target WAN interface Ethernet cable from the firewall.

For cable and DSL connections, try powering off the modem/CPE, and in a separate test, unplug the coax or phone line from the modem. For fiber, wireless, and other types of connections with a router outside of pfSense® software, try unplugging the Internet connection from the router, and also turning off the router itself.

All of the described testing scenarios will likely end with the same result. However, there are some circumstances where trying all these things individually will find a fault that would not have otherwise been noticed until an actual failure. One of the most common is unknowingly using a monitor IP address assigned to the DSL or cable modem. Hence when the coax or phone line is disconnected, simulating a provider failure rather than an Ethernet or modem failure, the monitor ping still succeeds since it is pinging the modem. From what the firewall was told to monitor, the connection is still up, so it will not fail over even if the upstream connection is actually down. There are other types of failure that can similarly only be detected by testing all the individual possibilities for failure. The monitor IP address can be edited on the gateway entry as covered in *Gateway Settings*.

Verifying Interface Status

After creating a WAN failure, refresh **Status > Gateways** to check the current status. As the gateway monitoring daemon notices the loss, the loss will eventually move past the configured alarm thresholds and it will mark the gateway as down.

18.9.2 Verifying Load Balancing Functionality

This section describes how to verify the functionality of a load balancing configuration.

Verifying HTTP Load Balancing

The easiest way to verify HTTP load balancing is to visit a website that displays the public IP address the client used to access the site. A page on the Netgate site is available for this purpose, and countless other sites offer the same functionality. Search for "what is my IP address" and numerous websites are returned that will show the public IP address making the HTTP request. Many of those sites tend to be full of advertisements, so Netgate provides a handful of sites which report only the client IP address:

- https://www.netgate.com/ip
- https://www.pfsense.org/ip
- https://files00.netgate.com/ip
- https://files01.netgate.com/ip

Browsers have a habit of keeping open server connections and caching results, so the best browser-based test is to either load multiple sites, or to close the browser window between attempts to load a site. During each connection attempt, a different IP address should be shown if load balancing is working correctly. If other traffic is present on the network, the IP address may not appear to change on every page load. Repeat the test several times and the IP address should change at least a few times. As an alternative to using a browser, command line utilities such as `curl` do not keep persistent sessions and are more accurate for testing this behavior.

If the IP address never changes, try several different sites, and make sure the browser is requesting the page again, and not returning something from its cache or using a persistent connection to the server. Other good steps include manually deleting the cache, closing and reopening the browser, and trying multiple web browsers. If all of those fail to return the expected result, then start troubleshooting the load balancer configuration further.

Testing load balancing with traceroute

The `traceroute` utility (or `tracert` in Windows) shows the network path taken to a given destination. See *Using traceroute* for details on using `traceroute`. With load balancing, running a `traceroute` from a client system behind the firewall should show a different path being taken for each attempt. Due to the way `traceroute` functions, wait at least one minute after stopping a traceroute before starting another test so that the states will expire, or try different destinations on each attempt.

Using Traffic Graphs

The real time traffic graphs under **Status > Traffic Graph** and on the Traffic Graphs dashboard widget are useful for showing the real time throughput on WAN interfaces. Only one graph at a time can be shown per browser window when using **Status > Traffic Graph**, but additional windows or tabs can be opened in the browser to see all WAN interfaces simultaneously. The traffic graphs widget for the Dashboard enables the simultaneous display of multiple traffic graphs on a single page to simplify this process. If load balancing is working correctly, activity will be observed on all WAN interfaces.

The RRD traffic graphs under **Status > Monitoring** are useful for longer-term and historical evaluation of WAN utilization.

Note: Bandwidth usage may not be exactly equally distributed, since connections are directed on a round robin basis without regard for bandwidth usage.

18.10 Multi-WAN on a Stick

In the router world, Cisco and others refer to a VLAN router as a "router on a stick" since it can be a functioning router with only one physical network connection. pfSense® software can be configured in this manner as well, using VLANs and a managed switch capable of 802.1q trunking. Most of the deployments running more than 5 WANs use this methodology to limit the number of physical interfaces required on the firewall. In such a deployment, the WAN interfaces all reside on one physical interface on the firewall, with the internal network(s) on additional physical interfaces. Figure *Multi-WAN on a stick* illustrates this type of deployment.

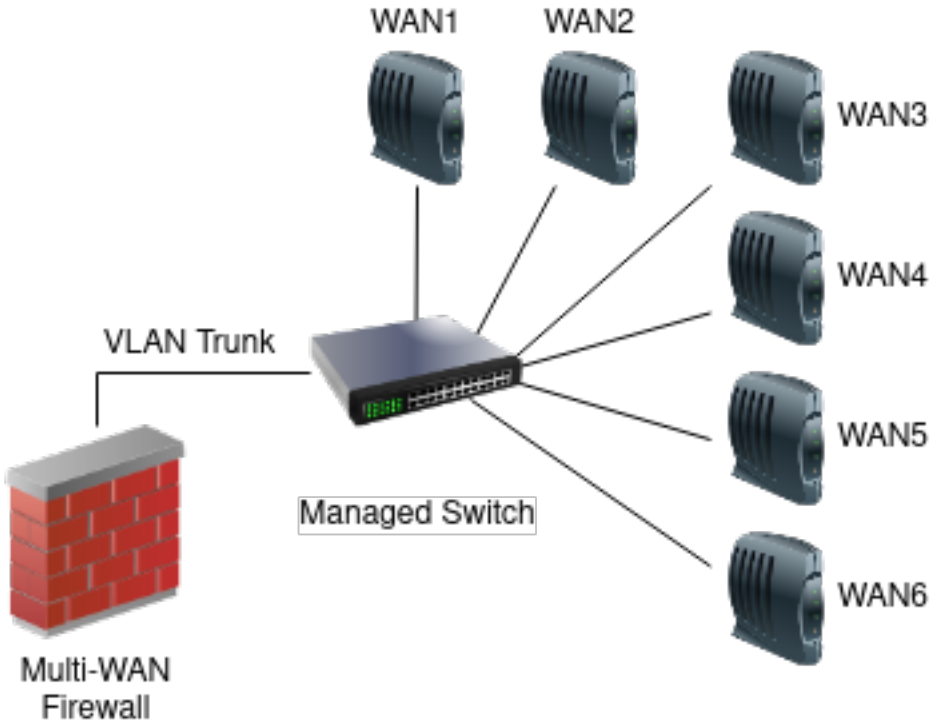


Fig. 2: Multi-WAN on a stick

18.11 Multi-Link PPPoE (MLPPP)

Multi-Link PPPoE (MLPPP) is a unique WAN option that bonds together multiple PPPoE lines from the same ISP to form one larger virtual circuit. This means a firewall can get the true aggregate bandwidth of all circuits in the bundle. For example, if a firewall has three 10 Mbit/s DSL lines in a bundle, it could potentially get 30Mbit/s from a single connection.

18.11.1 Requirements

The largest hurdle for MLPPP is that the ISP must support it on the circuits connected to the firewall. Few ISPs are willing to support MLPPP, so if an ISP is available that does, it would be worth taking advantage of that fact. Additionally, each line must be on a separate interface connected to the firewall running pfSense® software.

18.11.2 Setup

Setup for MLPPP is simple:

- Configure a WAN for a single line with the correct credentials
- Navigate to **Interfaces > Assign**, **PPPs** tab
- Click  to edit the entry for the PPPoE WAN
- Ctrl-click to select the other physical interfaces that belong to the same MLPPP bundle
- Click **Save**

The firewall will then attempt to bond the lines using MLPPP.

18.11.3 Caveats

One downside to using MLPPP is that troubleshooting is much more difficult. Statistics and status are not available for the individual lines. To determine the status, read through the PPP log, as there is not yet a way to query the lines separately. In some cases it's obvious if a line is down, as there may be a noticeable problem at the modem (out of sync) or that the maximum attainable bandwidth is reduced.

18.12 Using OpenVPN with Multi-WAN

OpenVPN servers can be used with any WAN, or multiple WANs, as can OpenVPN clients. This document covers only a remote access OpenVPN server, but a similar process could be applied for site to site VPNs.

There are many different ways to configure multiple WANs with OpenVPN on pfSense® software for remote access or site to site VPNs. Many of these were covered during the September 2014 pfSense Hangouts on Youtube.

See also:

The "Advanced OpenVPN Concepts" presentation is available through pfSense Hangouts on Youtube.

18.12.1 OpenVPN Configuration

First, get OpenVPN working as desired on the primary WAN interface. Once it is properly functioning, make a backup.

18.12.2 Bind to Localhost and Setup Port Forwards

The OpenVPN configuration needs to be adjusted so it can be reached from either WAN. The simplest way to do this is by changing the **Interface** on the VPN connection to be *Localhost*, and then adding a port forward on each WAN to redirect the OpenVPN port to *Localhost* (127.0.0.1).

For example: If there are two WANs and the OpenVPN server is running on port *1194*, set the **Interface** to *Localhost*, then add two port forwards:

- WAN1 - *UDP*, Source any, Destination *WAN1 Address* port `1194`, redirect target `127.0.0.1` port `1194`
- WAN2 - *UDP*, Source any, Destination *WAN2 Address* port `1194`, redirect target `127.0.0.1` port `1194`

18.12.3 Configure Clients

Clients may be configured to use the second WAN by adding a second *remote* statement to their configuration, such as:

```
remote x.x.x.x 1194 udp
```

Where `x.x.x.x` is the second WAN IP address or host name.

This process can be automated by using the OpenVPN Client Export package. When exporting a client, in **Host Name Resolution** choose one of:

Automagic Multi-WAN IPs (port forward targets) Adds a remote statement for each port forward found targeting the interface binding and port used by this VPN, uses the IP address of each WAN as-is.

Automagic Multi-WAN DDNS Hostnames (port forward targets) Like above, but uses the first located Dynamic DNS hostname for a given WAN. If the WAN is a private IP address, this may be the better choice.

18.12.4 More than two WAN connections

The same steps can be repeated to add more WAN connections. Add a port forward to any additional WAN. Clients will need an updated configuration file if another WAN is added later.

See also:

- *Troubleshooting Multi-WAN*
- *Configuring Multi-WAN for IPv6*

The multiple WAN (multi-WAN) capabilities in pfSense® software allow a firewall to utilize multiple Internet connections to achieve more reliable connectivity and greater throughput capacity.

Warning: Before proceeding with a multi-WAN configuration, the firewall must have a functional two interface (LAN and WAN) configuration.

pfSense software is capable of handling numerous WAN interfaces, with multiple deployments using over 10 WANs in production.

All WAN-type interfaces are treated identically in the GUI. Anything that can be done with the primary WAN can also be done with an additional OPT WAN interface. There are no significant differences between the primary WAN and additional WANs.

This section starts by covering items to consider when implementing *any* multi-WAN solution, then covers multi-WAN configuration with pfSense software.

18.13 Choosing Internet Connectivity

The ideal choice of Internet connectivity will depend largely upon the options available at a given location, but there are some additional factors to take into consideration.

18.13.1 Cable Paths

Speaking from the experience of those who have seen first hand the effects of multiple cable-seeking backhoes, as well as nefarious copper thieves, it is highly desirable to obtain connectivity choices for a multi-WAN deployment which utilize disparate cabling paths. In many locations, DSL connections as well as any others utilizing copper pairs are carried on a single cable subject to the same cable cut, and others from the same telco such as fiber circuits may run along the same poles or conduits.

If one connection comes in over copper pair (DSL), choose a secondary connection utilizing a different type and path of cabling. Cable connections are typically the most widely available option not subject to the same outage as copper services. Other options include fiber service or fixed wireless coming in on a different path from copper services.

Two connections of the same type cannot be relied upon to provide redundancy in most cases. An ISP outage or cable cut will commonly take down all connections of the same type. Some users use multiple DSL lines or multiple cable modems, though the only redundancy that typically offers is isolating a site from modem or other CPE (Customer Premise Equipment) failure. Consider multiple connections from the same provider as a solution only for additional bandwidth, as the redundancy such a deployment offers is minimal.

18.13.2 Paths to the Internet

Another consideration when selecting Internet connectivity for a site is the path from the connection itself to the Internet. For redundancy purposes, multiple Internet connections from the same provider, especially of the same type, should not be relied upon as they could all fail concurrently.

With larger providers, two different types of connections such as a Fiber line and DSL will usually traverse significantly different networks until reaching core parts of the network. These core network components are generally designed with high redundancy and any problems are addressed quickly since they have widespread effects. Hence such connectivity is isolated from most ISP issues, but since they commonly utilize the same cable path, it still leaves a site vulnerable to extended outages from cable cuts.

18.13.3 Better Redundancy, More Bandwidth, Less Money

In the past, high-grade telco services such as DS1 or DS3 circuits were the choice for environments with high availability requirements. Generally the Service Level Agreements (SLA) offered on DS1 and DS3 connections were better than other types of connectivity, and those circuits were generally seen as more reliable. End-users have largely left such circuits behind, however, because they are too slow or too costly by today's standards. With the multi-WAN capabilities on pfSense, a site can have more bandwidth and better redundancy for less money in many cases. Fiber services are rapidly becoming more widespread, shaking up this concept by providing extremely large amounts of bandwidth for relatively low cost, though such services may still have a less-than-desirable SLA for outage response.

Most organizations requiring high availability Internet connections do not want to rely upon DSL, cable or other "lesser class" broadband Internet connections. While they're usually significantly faster and cheaper, the lesser SLA is enough to make many companies think twice. In areas where multiple lower cost broadband options are available, such as fiber and cable, the combination of pfSense software and two low cost Internet connections provides more bandwidth and better redundancy at a lower cost. The chance of two different broadband connections going down simultaneously is significantly less than the chance of any single service outage. Adding a backup Cable or DSL line to supplement a much faster fiber line ensures connectivity will continue when an outage occurs on the fiber line, even if it is a rare occurrence.

CHAPTER

NINETEEN

VIRTUAL PRIVATE NETWORKS

19.1 Common deployments

There are four common uses of the VPN capabilities of pfSense, each covered in this section.

19.1.1 Site-to-site connectivity

Site-to-site connectivity is primarily used to connect networks in multiple physical locations where a dedicated, always-on, connection between the locations is required. This is frequently used to connect branch offices to a main office, connect the networks of business partners, or connect a network to another location such as a data center environment.

Before the proliferation of VPN technology, private WAN circuits were the only solution to connect multiple locations. These technologies include point-to-point dedicated circuits, packet switching technologies such as frame relay and ATM, and more recently, MPLS (Multiprotocol Label Switching) and fiber and copper based metropolitan Ethernet services. While these types of private WAN connectivity provide reliable, low latency connections, they are also very costly with recurring monthly fees. VPN technology has grown in popularity because it provides the same secure site to site connectivity using Internet connections that are generally much less costly.

Limitations of VPN connectivity

Performance is an important consideration when planning a VPN solution. In some networks, only a private WAN circuit can meet the requirements for bandwidth or latency. Latency is usually the biggest factor. A point to point DS1 circuit has end to end latency of about 3-5 ms, while the latency to the first hop on an ISP network will generally be at least that much if not higher. Metro Ethernet services or fiber circuits have end to end latency of about 0-3 ms, usually less than the latency to the first hop of an ISP network. That will vary some based on geographical distance between the sites. The stated numbers are typical for sites within a couple hundred miles of each other. VPNs usually see latency of around 30-60 ms depending on the Internet connections in use and the geographical distance between the locations. Latency can be minimized and VPN performance maximized by using the same ISP for all VPN locations, but this isn't always feasible.

Certain protocols perform very poorly with the latency inherent in connections over the Internet. Microsoft file sharing (SMB) is a common example. At sub-10 ms latency, it performs well. At 30 ms or higher, it's sluggish, and at more than 50 ms it's painfully slow, causing frequent hangs when browsing folders, saving files, etc. Getting a simple directory listing requires numerous round trip connections between the client and server, which significantly exacerbates the increased delay of the connection. In Windows Vista and Server 2008, Microsoft introduced SMB 2.0 which includes new capabilities to address the issue described here. SMB 2.0 enables the sending of multiple actions in a single request, as well as the ability to pipeline requests, meaning the client can send additional requests without waiting for the response from prior requests. If a network uses exclusively Vista and Server 2008 or newer operating systems this won't be a concern, but given the rarity of such environments, this will usually be a consideration. SMB 3.0 further improves in this area with support for multiple streams.

Two more examples of latency sensitive protocols are Microsoft Remote Desktop Protocol (RDP) and Citrix ICA. There is a clear performance and responsiveness difference with these protocols between sub-20 ms response times typically found in a private WAN, and the 50-60+ ms response times common to VPN connections. If remote users work on published desktops using thin client devices, there will be a notable performance difference between a private WAN and VPN. Whether that performance difference is significant enough to justify the expense of a private WAN will vary from one environment to another.

There may be other network applications in an environment that are latency sensitive, where the reduced performance of a VPN is unacceptable. Or all locations may be within a relatively small geographical area using the same ISP, where the performance of a VPN rivals that of private WAN connections.

19.1.2 Remote access

Remote access VPNs enable users to securely connect into a network from any location where an Internet connection is available. This is most frequently used for mobile workers (often referred to as "Road Warriors") whose job requires frequent travel and little time in the office, and to give employees the ability to work from home. It can also allow contractors or vendors temporary access to a network. With the proliferation of smart phones, users have a need to securely access internal services from their phones using a remote access VPN.

19.1.3 Protection for wireless networks

A VPN can provide an additional layer of protection for wireless networks. This protection is two-fold: It provides an additional layer of encryption for traffic traversing the wireless network, and it can be deployed in such a way that it requires additional authentication before access to network resources is permitted. This is deployed mostly the same as remote access VPNs. This is covered in *Additional protection for a wireless network*.

19.1.4 Secure relay

Remote access VPNs can be configured in a way that passes all traffic from the client system over the VPN. This is nice to have when using untrusted networks, such as wireless hotspots as it lets a client push all its Internet traffic over the VPN and out to the Internet from the VPN server. This protects the user from a number of attacks that are possible on untrusted networks, though it does have a performance impact since it adds additional hops and latency to all connections. That impact is usually minimal with high speed connectivity when the client and VPN server are relatively close geographically.

19.2 Choosing a VPN solution

Each VPN solution has pros and cons. This section will cover the primary considerations in choosing a VPN solution, providing the information necessary to choose the best solution for a given environment.

19.2.1 Interoperability

To interoperate with a firewall or router product from another vendor, the choices are limited by items supported by both sides.

Note: Interoperability in this sense isn't applicable with VPN types not listed here since they are not intended for site-to-site applications.

IPsec

IPsec is usually the best choice since it is included with nearly every VPN-capable device. It also prevents being locked into any particular firewall or VPN solution. For interoperable site-to-site connectivity, IPsec is usually the only choice.

OpenVPN

OpenVPN is interoperable with a few other packaged firewall/VPN solutions, but not many.

WireGuard

WireGuard, like OpenVPN, is compatible with only a few other packaged firewall/VPN solutions. However, as it is a more recently developed protocol, support is even more rare.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.2.2 Authentication considerations

All VPN types on the firewall support user authentication except for WireGuard. IPsec and OpenVPN can also work with shared keys or certificates. OpenVPN is a bit more flexible in this regard because it can work with only certificates, only shared keys, only user authentication, or a combination of these. Using OpenVPN with certificates, TLS authentication, and User Authentication is the most secure method. OpenVPN certificates can also be password protected, in which case a compromised certificate alone isn't adequate for connecting to a VPN if it is set to only use certificates. The lack of additional authentication can be a security risk in that a lost, stolen, or compromised system containing a key or certificate means whoever has access to the device can connect to a VPN until that loss is discovered and the certificate revoked.

While not ideal, a lack of username and password authentication on a VPN isn't as great a risk as it may seem. A compromised system can easily have a key logger installed to capture the username and password information and easily defeat that protection. In the case of lost or stolen systems containing keys, if the hard drive isn't encrypted, the keys can be used to connect. However adding password authentication isn't of great help there either, as usually the same username and password will be used to log into the computer, and most passwords are crackable within minutes using modern hardware when an attacker has access to an unencrypted drive. Password security is also frequently

compromised by users with notes on their laptop or in their laptop case with their password written down. As with any security implementation, the more layers utilized, the better, but it's always a good idea to keep these layers in perspective.

19.2.3 Ease of configuration

None of the available VPN options are extremely difficult to configure, but there are differences between the options.

IPsec

Has numerous configuration options and can be difficult for the uninitiated.

OpenVPN

OpenVPN requires the use of certificates for remote access in most environments, which comes with its own learning curve and can be a bit arduous to manage. There is a wizard to handle the most common OpenVPN remote access configurations and the OpenVPN client export packages eases the process of getting the clients up and running.

WireGuard

Has few options, thus configuration is simple. Lacks facilities for automated configuration and deployment leading to increased manual management.

19.2.4 Multi-WAN capable

If users require the ability to connect to multiple WANs, IPsec, OpenVPN, and WireGuard are capable of handling such configurations.

19.2.5 Client availability

VPN Client software is a program that handles connecting to the VPN and handling any other related tasks like authentication, encrypting, routing, etc. For remote access VPNs, the availability of VPN client software is a primary consideration. All options are cross platform compatible with many different operating systems but some require installing third-party clients. IPsec in EAP-MSCHAPv2 mode, IPsec in EAP-TLS mode, and IPsec in Xauth mode are the only options with client support built into some popular desktop and mobile operating systems. Other operating systems vary and may include more or less IPsec modes or may even include OpenVPN or WireGuard, as is the case with many Linux distributions. If using built-in clients is a must, consult the operating system documentation for all required client platforms to see if a common option is available and then check to see if that mode is possible.

In some cases multiple remote access VPNs may be required to accommodate all clients. For example, IPsec could be used for some and OpenVPN for others. Some organizations prefer to keep things consistent, so there is a trade-off to be made but for the sake of compatibility it may be worth offering multiple options.

IPsec

IPsec clients are available for Windows, Mac OS X, BSD, Linux, and others. Though the native clients may only support certain specific modes and configurations. General-use IPsec clients are not included in the OS except for some Linux and BSD distributions. Mac OS X includes both IKEv2 and Cisco (xauth) IPsec support. There are free and commercial options available with a user-friendly GUI.

OSX 10.11, along with Windows 7 and later include support for IPsec in specific modes using IKEv2: EAP-TLS and EAP-MSCHAPv2. Both options are supported and are covered in *Mobile IPsec*.

The Cisco-style IPsec client included with OS X and iOS devices is fully compatible with IPsec using xauth. Configuration for the iOS client is covered in *Configuring IPsec IKEv2 Remote Access VPN Clients on iOS*.

Many Android phones also include a compatible IPsec client, which is discussed in *Configuring IPsec IKEv2 Remote Access VPN Clients on Android*.

OpenVPN

OpenVPN has clients available for Windows, Mac OS X, all the BSDs, Linux, Solaris, and Windows Mobile, but the client does not come pre-installed in any of these operating systems.

Android devices can use a freely available OpenVPN client that works well and doesn't require rooting the device. That client is covered in *Installing the OpenVPN Client on Android*. There are other options available if the device is rooted, but that is beyond the scope of this documentation.

iOS also has a native OpenVPN client. For more information, see *Installing the OpenVPN Client on iOS*.

WireGuard

WireGuard clients are available for a variety of operating systems including Windows, Mac OS X, FreeBSD, Linux, Android, iOS, and more. Some Linux installations include WireGuard but in most cases it requires installation of a third party client.

19.2.6 Firewall friendliness

VPN protocols can cause difficulties for many firewalls and NAT devices. This is primarily relevant to remote access connectivity, where users will be behind a myriad of firewalls mostly controlled by third parties with varying configurations and capabilities.

IPsec

IPsec uses both UDP port 500 and the ESP protocol to function. Some firewalls don't handle ESP traffic well where NAT is involved, because the protocol does not have port numbers like TCP and UDP that make it easily trackable by NAT devices. IPsec clients behind NAT may require NAT Traversal to function, which encapsulates the ESP traffic over UDP port 4500.

OpenVPN

OpenVPN is very firewall-friendly. Since it uses a single UDP or TCP port and is not affected by common NAT functions such as rewriting of source ports, it is rare to find a firewall which will not work with OpenVPN. The only possible difficulty is if the protocol and port in use is blocked. Some administrators use a common port like UDP 53 (usually DNS), or TCP 80 (usually HTTP) or TCP 443 (usually HTTPS) or to evade most egress filtering.

WireGuard

Similar to OpenVPN in this regard, WireGuard uses a single UDP port and thus is not affected by firewall and NAT issues which may affect other protocols.

19.2.7 Cryptographically secure

One of the critical functions of a VPN is to ensure the confidentiality of the data transmitted.

IPsec

Tunnels using pre-shared keys can be broken if a weak key is used. Use a strong key, at least 10 characters in length containing a mix of upper and lowercase letters, numbers and symbols. Use of certificates is preferred, though somewhat more complicated to implement.

OpenVPN

Encryption is compromised if the PKI or shared keys are disclosed, though the use of multiple factors such as TLS authentication on top of PKI can mitigate some of the danger.

WireGuard

WireGuard encryption relies on pre-generated public/private key pairs and an optional pre-shared key. The peers only need the public keys of other peers, and the optional pre-shared key. The private keys and pre-shared key (if present) must be protected.

19.2.8 Support for NAT inside tunnels

While any use of NAT is undesirable, there are some occasions which can benefit from its use inside tunnels. Primarily, it can be useful for working around subnet conflicts or for setting up “outbound” style NAT when the remote endpoint only expects a single address (e.g. a VPN provider with no LAN-to-LAN routing)

IPsec

Support for NAT with IPsec depends on the mode, either tunnel or VTI.

Tunnel

Phase 2 entries in *Tunnel* mode support BINAT (1:1) and Overload/PAT style NAT. See *NAT with IPsec Phase 2 Networks* for details.

VTI

Phase 2 entries in *VTI* mode do not currently support NAT. See #8686 for details.

OpenVPN

OpenVPN supports inbound (e.g. port forwards) and outbound NAT using the group OpenVPN tab and also on assigned interfaces. Depending on the environment and configuration there may be some special considerations, such as ensuring proper return routing for post-NAT subnets.

WireGuard

WireGuard supports inbound (e.g. port forwards) and outbound NAT using the group WireGuard tab and also on assigned interfaces. Some cases may require using single peer tunnels or carefully crafted **Allowed IPs** lists to ensure correct return routing. See *Design Considerations* and *WireGuard and Rules / NAT*.

19.2.9 Per-tunnel Firewall Rules

Each VPN type has a common group tab for rules, and some also support rules for individual tunnels.

> **Warning:** Rules on group tabs are considered before per-interface rules. For per-interface rules to match, rules on the group tab must not match the same packets.

IPsec

IPsec does not currently support per-tunnel rules, its traffic can only be filtered by rules on the IPsec tab.

Even though phase 2 entries in *VTI* mode have an interface which can be assigned, they do not currently support interface rules. See #8686 for details.

OpenVPN

When assigned as an interface, OpenVPN instances fully support per-tunnel rules. See *Assigning OpenVPN Interfaces*.

WireGuard

When assigned as an interface, WireGuard instances fully support per-tunnel rules. See *Assign a WireGuard Interface* and *WireGuard and Rules / NAT*.

19.2.10 Automatic Mobile Client Configuration

Depending on the deployment, mobile (Remote Access) clients can receive automatic configuration in certain cases.

IPsec

In IKEv2 mode, clients can automatically receive an IP address allocated from a pool, along with DNS configuration.

In IKEv1 mode with Xauth, in addition to the above, clients can also receive a list of networks to route across the VPN.

OpenVPN

OpenVPN clients can automatically receive an IP address allocated from a pool, and numerous additional options can be pushed to clients to control their behavior from the server side (routing, DNS, and many others).

WireGuard

WireGuard mobile clients must be configured statically. On the server side, a client tunnel address must be setup in the **Allowed IPs** for a peer. The same address must be configured on the client. This varies by OS/Platform, some read it from the configuration and other require it to be configured on interfaces via CLI commands. Networks to route must likewise be manually added on the client configuration **Allowed IPs** list and, depending on the client, may also need to be added to its operating system routing table.

19.2.11 Routing Support

IPsec

IPsec in *Tunnel* mode uses policies, not routes, and thus does not respect the operating system routing table.

IPsec in *VTI* mode supports static and dynamic routing (e.g. BGP, OSPF) and works with the operating system routing table.

OpenVPN

In SSL/TLS tun mode with multiple clients, OpenVPN uses its internal routing on client-specific configurations to determine which clients receive traffic for specific subnets. In this type of configuration, dynamic routing is not possible.

In SSL/TLS tun mode with a /30 subnet (one client per server) or in Shared Key mode, dynamic routing is possible using OSPF or BGP. This is due to the fact that in these cases, OpenVPN does not need to track internal routing and can rely on the operating system routing table alone.

In tap mode, dynamic routing is possible as packets can be handed off using L2/ARP information rather than relying on internal routing in OpenVPN.

WireGuard

WireGuard routes specific subnets to peers based on the **Allowed IPs** list, but also requires operating system routing table entries for traffic to enter a WireGuard tunnel.

When a WireGuard tunnel has more than one peer, the **Allowed IPs** list lets WireGuard determine internally which clients receive traffic for specific subnets. Due to this internal routing, dynamic routing is not possible in a configuration where WireGuard has multiple peers per tunnel.

For a WireGuard tunnel with a single peer, WireGuard can forward arbitrary networks to the peer without having them all listed in **Allowed IPs**. Thus, in this situation, it can take advantage of dynamic routing using BGP. OSPF is also possible but requires additional configuration.

See *WireGuard Routing* for more information.

19.2.12 Recap

Table *Features and Characteristics by VPN Type* shows an overview of the considerations provided in this section.

Table 1: Features and Characteristics by VPN Type

VPN Feature	IPsec	OpenVPN	WireGuard
User Authentication	Yes	Yes	No
Client included in most OSes	Varies by mode	No	No
Widely interoperable	Yes	No	No
Multi-WAN	Yes	Yes	Yes
Cryptographically secure	Yes	Yes	Yes
Firewall friendly	Yes (NAT-T or IKEv2)	Yes	Yes
In-tunnel NAT	Limited	Yes	Yes
Per-tunnel Firewall Rules	No	Yes	Yes
Automatic Mobile Config	Yes	Yes	No
Static Routing	Yes (VTI Only)	Internal	Internal
Dynamic Routing	Yes (VTI Only)	Varies	Varies

19.3 Remote Access Mobile VPN Client Compatibility

A variety of remote access ("mobile") VPN configuration styles are available to accommodate nearly any potential client. The table below shows which operating systems have compatible clients with some of the most common remote access VPN configurations.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

Protocol/Operating System	Windows 10	Android	iOS	OS X
OpenVPN	*3PA* 1 2	*3PA* 4	*3PA* 6	*3PA* 2
WireGuard	3PA 3	3PA 3	3PA 3	3PA 3
IPsec PSK	?	Varies	?	?
IPsec RSA	?	Varies	?	?
IPsec IKEv2 EAP-MSCHAPv2/RADIUS	*Yes*	*3PA* 5	*Yes*	*Yes*
IPsec IKEv2 EAP-TLS	*Yes*	3PA 5	*Yes*	*Yes*

Table: Mobile/Remote Access VPN Client Availability

- Yes = OS Native Client Available
- 3PA = Third Party Client Required
- Varies = Varies by device model and vendor options

Unless otherwise stated, UNIX clients (*BSD, Linux, etc) can support any style with manual configurations but the availability of GUI configuration tools varies by distribution.

19.4 VPNs and Firewall Rules

VPNs and firewall rules are handled somewhat inconsistently in pfSense® software. This section describes how firewall rules are handled for each of the individual VPN options. For the automatically added rules discussed here, the addition of those rules may be disabled by checking **Disable all auto-added VPN rules** under **System > Advanced** on the **Firewall/NAT** tab.

19.4.1 IPsec

IPsec traffic coming in to the specified WAN interface is automatically allowed as described in *IPsec*.

Traffic encapsulated within an active IPsec connection is controlled via user-defined rules on the **IPsec** tab under **Firewall > Rules**.

19.4.2 OpenVPN

OpenVPN does not automatically add rules to WAN interfaces. The OpenVPN remote access VPN Wizard offers to optionally create rules to pass WAN traffic and traffic on the OpenVPN interface.

Traffic encapsulated within an active OpenVPN connection is controlled via user-defined rules on the **OpenVPN** tab under **Firewall > Rules**.

OpenVPN interfaces may also be assigned similar to other interfaces. In such cases the **OpenVPN** tab firewall rules still apply, but there is a separate tab specific to the assigned VPN instance that controls traffic only for that one VPN.

19.4.3 WireGuard

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

WireGuard does not automatically add rules to WAN interfaces. Rules must be added to the appropriate WAN interface(s) to allow traffic to reach the ports for WireGuard instances.

Traffic encapsulated within WireGuard is controlled via user-defined rules on the **WireGuard** tab under **Firewall > Rules**.

WireGuard interfaces may also be assigned similar to other interfaces. In such cases the **WireGuard** tab firewall rules still apply, but there is a separate tab specific to the assigned VPN instance that controls traffic only for that one VPN.

19.5 IPv6 VPN and Firewall Rules

As mentioned briefly in *Firewall and VPN Concerns*, special care must be taken when routing IPv6 traffic across a VPN and using publicly routable subnets. The same advice also applies to IPv4 but it's much less common to have clients on both sides of an IPv4 VPN using publicly routable addresses.

The main issue is that because it's possible to route all the way from one LAN to the other LAN across the Internet, then traffic could be flowing unencrypted between the two networks if the VPN is down (or not present at all!). This is far from ideal because although connectivity is available, if any traffic were intercepted in between the two networks and that traffic was using an unencrypted protocol like HTTP, then it could compromise the network.

One way to prevent this is to not allow traffic from the remote IPv6 LAN in on the opposing side's WAN rules. Only allow traffic from the remote side's subnet on the firewall rules for whichever VPN type is being used to protect the traffic. An explicit block rule could also be added to the top of the WAN rules to ensure that this traffic cannot enter from the WAN directly. A better method is to use a floating rule to reject outbound traffic on WAN destined for VPN hosts/remote local networks. This way the insecure traffic never leaves the premises. With the rule set to log, the "leakage" would be obvious to someone monitoring the logs as it would be shown blocked outbound on WAN.

Another less obvious consequence of having dual stack connectivity between networks is that differences in DNS can cause unintended routing to take place. Suppose IPv4 VPN connectivity exists between two sites, but there is no IPv6 VPN, only standard IPv6 connectivity at both locations. If a local host is set to prefer IPv6 and it receives a AAAA DNS response with the IPv6 IP address for a remote resource, it would attempt to connect over IPv6 first rather than using the VPN. In cases such as this, care would be needed to make sure that DNS does not contain conflicting records or that floating rules are added to prevent this IPv6 traffic from leaking out WAN. A more in-depth article on these kinds of traffic leakage can be found in the IETF draft named draft-gont-opsec-vpn-leakages-00.

19.6 VPN Scaling

The advice on this page is intended to help firewall administrators handle increased VPN volume, both in terms of throughput and number of connected users.

Warning: The advice on this page is relayed from experience and from community members. The advice on this page may not apply to all environments or use cases, and has not been definitively proven to help, but is offered in case others find it useful.

See also:

This document won't cover all factors of choosing between VPN types or how to setup VPNs, that information can be found *elsewhere in this documentation* and in pfSense Hangouts on Youtube.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.6.1 General Advice

No Artificial Limits

The firewall does not place any artificial limits on VPN connections. Any limitations encountered are due to settings, the hardware/environment, or the underlying technology.

IPsec is Fastest

IPsec is faster than other choices when configured correctly and when using hardware acceleration. If both client and server support IPsec, use IPsec. WireGuard is nearly as fast, however, and depending on the environment and availability of hardware acceleration, may be faster in some cases.

Use External Authentication

For user-based authentication, the most efficient method of user management for large numbers of accounts is an external authentication source, such as a RADIUS server, LDAP server, Active Directory (Via LDAP or RADIUS/NPS), etc.

Check Logs

If additional users are unable to connect, look in the logs on both the client and server side for specific error messages before seeking support.

Use Hardware Acceleration

Using a cryptographic accelerator such as a CPU with AES-NI will help greatly with throughput and crypto-related tasks. Enable the AES-NI and BSD `cryptodev` modules under **System > Advanced** on the **Miscellaneous** tab.

Use AES-GCM or ChaCha20-Poly1305

Using efficient encryption will increase security and performance. Authenticated Encryption with Additional Data (AEAD) ciphers combine encryption and authentication in one step, eliminating the need for additional hashing. AES-GCM and ChaCha20-Poly1305 are both AEAD ciphers. IPsec and OpenVPN can use AES-GCM. OpenVPN can also use ChaCha20-Poly1305, and that is also the only cipher supported by WireGuard.

Client support varies by platform.

Use Accelerated Ciphers

Certain hardware may accelerate ciphers so that choices are faster or more efficient. For hardware sold by Netgate, see the Netgate Appliances page for performance data and recommendations.

Disable Performance-Limiting Mitigation Settings

While we do not have any data on if or by how much they may impact VPNs, CPU vulnerability mitigation methods such as Kernel PTI and MDS mode can potentially degrade total performance. The potential for exploitation is minimal since arbitrary code cannot be run on the firewall except by users which already have the equivalent of administrator-level access. To ensure this risk stays low, only allow trusted administrators to access the firewall GUI and shell (SSH or console). The settings to enable/disable these features are under **System > Advanced** on the **Miscellaneous** tab.

Check Tunnel Network/Virtual Address Pool Sizes

Both IPsec and OpenVPN can assign addresses to clients out of a pool for remote access/mobile VPNs. The sizing of this pool limits how many clients can connect. For example, the maximum number of users in a /24 pool is 252, but other settings may reduce that value. See the sections below for more specific advice.

Use "Secure Enough" Settings

While we do not recommend deliberately using weak configurations, in some cases trade-offs are made for security between two secure ciphers or settings where one may offer *even better* security, but the lower of the two is still secure. In these cases, using the "Secure Enough" option can provide efficiency vs increased security. So long as the decision is informed, there may be some performance gained without compromising security in an unacceptable way. For example, with AES-GCM a key length of 128 bits is still considered secure. A 256 bit key is more secure, but the 256 bit key could put more of a burden on the hardware.

Consider Split Tunneling

Configurations which send all client data over the VPN, including Internet-bound traffic, will consume more resources than those which only send traffic for specific subnets. There are plenty of valid reasons to use either kind of configuration, however, when resources are stretched thin, easing the traffic burden on the VPN may justify switching to split tunneling rather than tunneling everything. Depending on the type of VPN and client, this may require adjustments on the server, the client, or both. See the sections below for specific recommendations.

Use Multiple Firewalls

In some instances, the burden may be too great for any single firewall to handle. In cases like this, multiple firewalls can be used to handle the required number of clients or throughput, at a cost of greatly increased complexity. There is no way to automatically balance between nodes in this manner, but such a configuration could be manually managed. This would also likely require the capability to have multiple external addresses on the WAN so each firewall can work in parallel, and also increases the complexity of routing on the internal side.

Use TNSR

TNSR® software is capable of vastly increased total IPsec throughput compared to pfSense® software. If pfSense software is unable to reach the throughput needs for a given use case, see the TNSR product page for more information.

19.6.2 Scaling IPsec

IPsec is well-suited to high throughput by default, especially given the advice above, but there are additional IPsec-specific tweaks which may help.

Note: See the TNSR product page for information about using TNSR for even larger total site-to-site throughput needs.

Optimal Encryption Settings

- Use AES-NI capable hardware.
- In Phase 1 (IKE) settings, use:
  - *AES128-GCM* with *128 bit* key length for the Algorithm
  - *AES-XCBC* for the hash, which in this case is effectively a Pseudo-Random Function (PRF).
- In Phase 2 (Child SA) settings, use:
  - *AES128-GCM* with *128 bit* key length for the Algorithm
  - **Do not select any Hash Algorithms**. A hash algorithm is unnecessary for AES-GCM as it already includes authentication.

Enable Multiple Phase 1 and Phase 2 Proposals

Multiple Phase 1 and Phase 2 encryption proposals may be configured in the GUI. Enabling multiple combinations of settings will allow peers to choose the most optimal settings which both sides support.

Enable Asynchronous Cryptography

IPsec cryptography jobs can be dispatched multi-threaded to run in parallel and increase performance. However, not all platforms and configurations fully support this function. To enable this capability, check **Asynchronous Cryptography** under **VPN > IPsec** on the **Advanced** tab.

> **Warning:** Be on the lookout for IPsec traffic drops/failures to pass with this setting enabled. See https://redmine.pfsense.org/issues/8964 for more information.

Split Tunneling

As mentioned above, split tunneling would only send traffic for specific subnets across the VPN rather than sending all traffic. On IPsec, this can be done in some cases by listing the specific networks in Phase 2 entries for the Mobile IPsec P1 rather than `0.0.0.0/0`. On the mobile clients tab, set **Provide a list of accessible networks to clients**. Even with that set, certain cases such as Windows 10 may require additional changes to direct clients to send only specific traffic over the tunnel.

19.6.3 Scaling OpenVPN

Use IPsec or WireGuard Instead

As mentioned previously, where possible, use IPsec or WireGuard instead. IPsec and WireGuard are much more efficiently integrated into the operating system, and are capable of much greater throughput than OpenVPN.

Use UDP

UDP has less overhead for tunneled data, and if a client has to retransmit, it won't compound the problem by retransmitting both inside and outside the tunnel. Unless there are extenuating circumstances which require TCP, use UDP.

Use TLS for Authentication Only

OpenVPN can use TLS for both authentication and for encryption of the control channel. Performing control channel encryption adds more overhead, which can add up with many clients. If control channel encryption is not required, consider using TLS for only authentication instead. No matter which option is chosen, traffic carried by OpenVPN is encrypted.

Encryption Algorithm

Use a CPU with AES-NI when possible, and use AES-GCM for the Encryption Algorithm when possible. ChaCha20-Poly1305 may also give a performance boost if hardware acceleration is unavailable. Note that for AEAD ciphers such as AES-GCM and ChaCha20-Poly1305, OpenVPN ignores the setting for **Auth Digest Algorithm**.

Note: For OpenVPN, AES-GCM and ChaCha20-Poly1305 can only be used in SSL/TLS mode, not Shared Key mode.

Use Data Cipher Negotiation

Data cipher negotiation can be used to set preferences so that more efficient ciphers can be preferred by clients where possible, but others can be used when necessary. Set high-priority selections such as *AES-128-GCM* first, followed by others like *AES-128-CBC*.

Split Tunneling

As mentioned in the general section above, split tunneling only sends traffic for specific subnets across the VPN rather than sending all traffic. With OpenVPN, this can be done by **Unchecking** the **Redirect IPv4/IPv6 Gateway** option(s) and configuring **IPv4/IPv6 Local Network(s)** entries instead. Clients may still override this behavior remotely, however, so check the client configurations as well.

Concurrent Connections

The firewall does not impose any connection limits by default, but an administrator may have chosen to configure a limit on the number of connections via the **Concurrent Connections** setting on servers. Ensure this is either unset or set high enough to accommodate the required number of users.

Disable Compression

Though using compression is tempting to squeeze extra throughput out of slower links, it is both inefficient and insecure. Most data sent across VPNs in modern environments is already encrypted or otherwise uncompressible, which wastes CPU when attempting to compress. Additionally, vulnerabilities such as VORACLE can allow attackers to glean information about encrypted data when it has been compressed. Disabling encryption will mitigate that attack and also reduce CPU overhead. On the server, set **Compression** to *Disable Compression*.

Duplicate Connections

Normally, if an OpenVPN client connects using the same username or certificate CN, the older connection is broken in favor of the new connection. This is more secure, but does not allow any given user to connect multiple times. Circumstances may necessitate supporting this, and in some environments it's not possible to give every device a unique username and/or certificate. Check **Duplicate Connection** in the OpenVPN server settings to allow multiple connections from the same user.

Topology

OpenVPN defaults to *subnet* topology which uses addresses more efficiently, but if the VPN was configured initially on older versions, or if an older guide was followed, it may still be using *net30* topology. Using a common example tunnel network of `10.0.8.0/24`, with *subnet* topology, the VPN can have a maximum of 252 users but with *net30*, it can only have 63. This is because in *net30* mode, each user receives a /30 subnet which utilizes four IP addresses for each user. In *subnet* mode, the server uses a single address and the client uses a single address, which is much more efficient.

Use UDP Fast I/O

This option is experimental but for those who have used it, it can result in much higher throughput. Not all platforms support it, however.

Increase Send/Receive Buffer

The default buffer size is safe, but not optimal. Increasing the buffer size to 512KiB on both sides can result in greater throughput. Results will vary by platform, internet link speed, and other factors. May require experimenting with multiple values to find the most efficient setting for a given environment.

Use Multiple Servers

OpenVPN is not multi-threaded so any single instance of OpenVPN is limited to using a single CPU. If a router has fast cores and not too many users, that may be OK, but it does not scale well. A workaround for this is to split users onto multiple servers. There are various means to reach this goal, including (but not limited to):

- Multiple servers on different WANs or ports, each with unique tunnel networks but otherwise identical settings (Same CA structure, encryption, etc).
  - Administrators could choose to manually configure pools of clients to connect to specific servers, but that does not scale well.
  - Clients may connect to any server configured in this manner so long as their settings line up properly.
  - Multiple servers can be listed in a single client configuration with additional `remote` statements.
  - Add `remote-random` to the client configuration so that clients will pick a random server when starting, which avoids overloading whichever server is listed first.
  - Servers could be run on multiple WANs to overcome single-circuit throughput limits.
- Multiple servers with completely unique settings (Different CA structure, different clients, etc)
  - More secure but more difficult to manage.
  - Clients must use different configurations to reach each server, no automated/built-in way to pick between them unless a specific client supports that function.
  - Good for isolating separate security levels (e.g. remote workers, remote administrators, vendors).

Process Efficiency

As a counterpoint to the above, each server will incur additional memory and other overhead to manage the process. When dealing with site-to-site VPNs, it is more efficient from a *memory* standpoint to use a single server with multiple clients (Peer to Peer SSL/TLS) vs servers for every node (Peer to Peer Shared Key, or SSL/TLS with a /30 tunnel network). If memory is a limiting factor, use fewer servers. If CPU overhead is the limiting factor, use separate servers.

19.7 OpenVPN

19.7.1 OpenVPN and IPv6

OpenVPN can connect a site-to-site tunnel to either an IPv4 address or an IPv6 address and both IPv4 and IPv6 traffic may be passed inside of an OpenVPN tunnel at the same time. IPv6 is supported both in site-to-site and mobile clients, and it can be used to deliver IPv6 to a site that only has IPv4 connectivity. In order to ensure mobile client support for IPv6, obtain the client software from the OpenVPN client export package, or download a client based on OpenVPN 2.3 or newer.

19.7.2 OpenVPN Configuration Options

This section describes all of the available options with OpenVPN and when they are typically used. Subsequent sections cover examples of configuring site-to-site and remote access VPNs with OpenVPN, using the most common options and a minimal configuration.

Server Configuration Options

These options are available in one or more modes for OpenVPN server instances, managed from **VPN > OpenVPN**, on the **Servers** tab.

Disable this server

Check this box and click **Save** to retain the configuration, but not enable the server. The process for this instance will be stopped, and all peers/clients will be disconnected from this server. Any other active servers are unaffected.

Server Mode

This is the role for the server, which specifies how routers or users will connect to this server instance. Changing this will also affect what options will appear on the rest of the page, so only relevant choices are displayed.

Peer to Peer (SSL/TLS) A connection between local and remote networks that is secured by SSL/TLS. This choice offers increased security as well as the ability for the server to push configuration commands to the remote peer router when using a 1:many style setup. Remote peer routers can also have certificates revoked to remove access if they become compromised.

Peer to Peer (Shared Key) A connection between local and remote networks that is secured by a single Shared Key configured on both nodes. This choice is easier to setup, but is less secure. If a shared key is compromised, a new key must be generated and then copied to any router or client using the old shared key. In this mode, a separate server instance is needed for each client.

Remote Access (SSL/TLS) This choice is a mobile client setup with per-user X.509 certificates. As with the peer-to-peer SSL/TLS connection type, using this method offers increased security as well as the ability for the server to push configuration commands to clients. Mobile clients can also have keys revoked to remove access if a key is compromised, such as a stolen or misplaced laptop.

Remote Access (User Auth) A client access server that does not use certificates, but does require the end user to supply a username and password when making a connection. This is not recommended unless authentication is handled externally by LDAP or RADIUS.

Remote Access (SSL/TLS + User Auth) The most secure choice offered. Not only does it get the benefits of other SSL/TLS choices, but it also requires a username and password from the client when it connects. Client access can be removed not only by revoking the certificate, but also by changing the password. Also, if a compromised key is not immediately discovered, the danger is lessened because it is unlikely that the attacker has the keys and the password. When using the OpenVPN wizard, this is the mode which is configured during that process.

Protocol

TCP or UDP may be selected, or their IPv6-enabled counterparts, TCP6 or UDP6. An OpenVPN server instance can currently only bind to either IPv4 or IPv6, but not both at the same time. UDP is the most reliable and fastest choice for running OpenVPN, and it should always be used when possible. In some rare cases TCP can be used to work around limitations, such as bypassing some firewalls by running an OpenVPN server on TCP port 443.

Connectionless protocols such as UDP are always preferable when tunneling traffic. TCP is connection oriented with guaranteed delivery, so any lost packets are retransmitted. This sounds like a good idea on the surface but TCP retransmissions will cause performance to degrade significantly on heavily loaded Internet connections or those with consistent packet loss.

TCP traffic frequently exists within tunnels and it is undesirable to retransmit lost packets of encapsulated VPN traffic. In cases where TCP is wrapped around TCP, such as a VPN tunnel using TCP as a transport protocol, when a packet is lost both the outer and inner lost TCP packets will be re-transmitted. Infrequent occurrences of this will be unnoticeable but recurring loss will cause significantly lower performance than UDP. If the traffic inside the tunnel requires reliable delivery, it will be using a protocol such as TCP which ensures that and will handle its own retransmissions.

Device Mode

OpenVPN can run in one of two device modes: *tun* or *tap*:

tun Works on OSI layer 3 and performs routing on point-to-point interfaces.

tap Can work at OSI layer 2 and can perform both routing and bridging if necessary.

Note: Not all clients support tap mode, using tun is more stable and more widely supported. Specifically, clients such as those found on Android and iOS only support tun mode in the Apps most people can use. Some Android and iOS OpenVPN apps that require rooting or jailbreaking a device do support tap, but the consequences of doing so can be a bit too high for most users.

Interface

Selects the interface, VIP, or failover group that the OpenVPN server instance will listen upon for incoming connections. This also controls which interface the traffic from the server will exit.

Several types of options are listed in the drop-down for **Interface**, and some have special behavior or use cases:

Interfaces OpenVPN will bind to the interface address. If the interface is dynamic, such as DHCP, OpenVPN will automatically bind to the new address if it changes.

VIPs OpenVPN will bind only to the specified VIP (IP Alias or CARP type)

Gateway Groups For use with failover groups, OpenVPN will bind to the address of the interface that is currently active in the group. If that interface gateway becomes unreachable, the next one will be used instead, and so on.

Localhost Useful for Multi-WAN deployments, binding to localhost and utilizing port forwards to accept connections from several interfaces and/or ports is a versatile way to provide redundant OpenVPN connectivity for connecting clients.

Any Binds to every address on every interface. Though tempting, this option is not recommended. When used with UDP, replies to Internet clients will always exit back out the default gateway WAN, which may be undesirable.

Local port

The local port is the port number OpenVPN will use to listen. Firewall rules need to allow traffic to this port and it must be specified in the client configuration. The port for each server must be unique for each interface.

Description

Enter a description for this server configuration, for reference.

Cryptographic Settings

This section controls how traffic to and from clients is encrypted and validated.

Shared Key

When using a shared key instance, either check the **Automatically generate a shared key** box to make a new key, or uncheck the box to paste in a shared key from an existing OpenVPN tunnel. When generating the key automatically, return to the edit screen for this tunnel later to obtain the key which may be copied to the remote router.

TLS Authentication

TLS, or Transport Layer Security, provides session authentication to ensure the validity of both the client and the server. Check the box to **Enable authentication of TLS packets** if desired. If there is no existing TLS key, leave **Automatically generate a shared TLS authentication key** checked. If key already exists, uncheck that option and then paste it into the provided entry box. When generating the key automatically, return to the edit screen for this tunnel later to obtain the key which may be copied to the remote router or client.

Warning: When using an SSL/TLS mode, we strongly recommend using TLS Authentication as well. In addition to the added security benefit from the key requirement, a TLS key also helps protect against some SSL-based attacks such as Heartbleed.

Peer Certificate Authority

Select the certificate authority used to sign the client or peer certificate(s) for this OpenVPN server instance. If none appear in this list, first import or generate a certificate authority under **System > Cert Manager**, on the **CAs** tab.

Peer Certificate Revocation List

This optional field is for the Certificate Revocation List (CRL) to be used by this tunnel. A CRL is a list of certificates made from a given CA that are no longer considered valid. This could be due to a certificate being compromised or lost, such as from a stolen laptop, spyware infection, etc. A CRL can be created or managed from **System > Cert Manager**, on the **Certificate Revocation** tab.

Server Certificate

A server certificate must be chosen for each OpenVPN server instance. If none appear in this list, first import or generate a certificate authority under **System > Cert Manager**, on the **Certificates** tab.

DH Parameters Length

The Diffie-Hellman (DH) key exchange parameters are used for establishing a secure communications channel. They may be regenerated at any time, and are not specific to an OpenVPN instance. That is, when importing an existing OpenVPN configuration these parameters do not need to be copied from the previous server. The length of the desired DH parameters may be chosen from the drop-down box, either 1024, 2048, or 4096.

Note: Due to the heavy computation involved in generating DH keys, a pre-generated set for each key type is used. New DH parameters may be generated manually by using the following shell commands:

```
# /usr/bin/openssl dhparam 1024 > /etc/dh-parameters.1024
# /usr/bin/openssl dhparam 2048 > /etc/dh-parameters.2048
# /usr/bin/openssl dhparam 4096 > /etc/dh-parameters.4096
```

Encryption algorithm

The cryptographic cipher to be used for this connection. The default is *AES- 128-CBC*, which is AES 128 bit Cipher Block Chaining. This is a fine choice for most scenarios.

See also:

Hardware Crypto for more information on using cryptographic accelerators and choosing an encryption algorithm.

Auth Digest Algorithm

Selects the message digest algorithm to use for HMAC authentication of incoming packets.

Note: OpenVPN defaults to SHA1 when this option is not specified, so unless both sides are set to a known value, use SHA1 here.

Hardware Crypto

If available, this option controls which hardware cryptographic accelerator will be used by OpenVPN. When left unspecified, OpenVPN will choose automatically based on what is available in the Operating System.

If this firewall device has a hardware cryptographic accelerator, choose **BSD Cryptodev Engine**, or select the specific device if it appears in the list. Most accelerator boards use the BSD cryptodev engine, so when in doubt, select that. This setting will allow OpenVPN to take advantage of the hardware acceleration. An encryption algorithm supported by the accelerator must also be selected. Refer to the hardware documentation for information on ciphers supported by the accelerator.

Certificate Depth

This option limits the length of a certificate chain before it fails validation. This defaults to *One (Client+Server)* so that if somehow an unauthorized intermediate CA is generated, certificates signed by the rogue intermediate would fail validation. In cases when chaining with intermediates is required, this limit can be raised.

Strict User-CN Matching

For SSL/TLS+User Authentication server, when enabled, this option enforces a match between the username supplied by the user and the Common Name of their user certificate. If the two do not match, the connection is rejected. This prevents users from using their own credentials with another person's certificate and vice versa.

Tunnel Settings

The tunnel settings section governs how traffic flows between the server and clients, including routing and compression.

IPv4/IPv6 Tunnel Network

These are the pools of addresses to be assigned to clients upon connecting. The server's end of the OpenVPN configuration will use the first address in this pool for its end of the connection, and assign additional addresses to connected clients as needed. These addresses are used for direct communication between tunnel endpoints, even when connecting two existing remote networks. Any subnet may be chosen provided that it is not in use locally or at any remote site. One or both of **IPv4 Tunnel Network** and **IPv6 Tunnel Network** may be entered, or in the case of a tap bridge, neither.

> **Warning:** Currently, limitations in OpenVPN itself prevent running with only an **IPv6 Tunnel Network** configured. When an **IPv6 Tunnel network** is defined, an **IPv4 Tunnel Network** must also be specified, even if it is not used.

For a site-to-site SSL/TLS server using IPv4, the **IPv4 Tunnel Network** size can alter how the server behaves. If `x.x.x.x/30` is entered for the **IPv4 Tunnel Network** then the server will use a peer-to-peer mode much like Shared Key operates: It can only have one client, does not require client-specific overrides or *iroutes*, but also cannot push routes or settings to clients. If an **IPv4 Tunnel Network** larger than that is used, such as `x.x.x.x/24`, the server will accept multiple clients and can push settings, but does require *iroutes*.

See also:

See *OpenVPN Site-to-Site Configuration Example with SSL/TLS* for more information on a site-to-multi-site example using a large tunnel network and *iroutes*.

Bridging Options

When using *tap* mode, additional options are shown that control bridging behavior in OpenVPN and client address assignment. These are covered in *Bridging OpenVPN Connections to Local Networks*

Redirect Gateway

When the **Redirect Gateway** option is selected the server will push a message to clients instructing them to forward *all* traffic, including Internet traffic, over the VPN tunnel. This only works in SSL/TLS modes with a tunnel network larger than a /30 subnet.

IPv4/IPv6 Local network

These fields specify which local networks are reachable by VPN clients, if any. A route for these networks is pushed to clients connecting to this server. If multiple routes for subnets of a particular family are needed, enter the subnets separated by a comma, e.g. `192.168.2.0/24, 192.168.56.0/24`.

This function relies upon the ability to push routes to the client, so for IPv4 it is only valid in an SSL/TLS context when a tunnel network larger than a `/30` is in use. It will always work for IPv6 provided a similar too-small mask isn't set.

IPv4/IPv6 Remote Network

This option only appears when a Peer-to-Peer type connection is used, and is not available for mobile clients. Routes table entries are added to the firewall for the specified subnets, which hand the traffic over to this OpenVPN instance for processing. If more than one Remote network subnet is needed, enter the subnets separated by a comma, e.g. `192.168.2.0/24, 192.168.56.0/24`.

Concurrent Connections

Specifies the number of clients that may be simultaneously connected to this OpenVPN server instance at any given time. This is a collective limit for all connected clients, not a per-user setting.

Compression

When compression is enabled, traffic crossing the OpenVPN connection will be compressed before being encrypted. This saves on bandwidth usage for many types of traffic at the expense of increased CPU utilization on both the server and client. Generally this impact is minimal, and enabling compression is beneficial for nearly any usage of OpenVPN over the Internet.

For high speed connections, such as the usage of OpenVPN across a LAN, high speed low/latency WAN, or local wireless network, this may be undesirable, as the delay added by the compression may be more than the delay saved in transmitting the traffic. If nearly all of the traffic crossing the OpenVPN connection is already encrypted (such as SSH, SCP, HTTPS, among many other protocols), do not enable LZO compression because encrypted data is not compressible and the LZO compression will cause slightly more data to be transferred than would be without compression. The same is true if the VPN traffic is almost entirely data that is already compressed.

This selector controls the handling of LZO compression for this OpenVPN instance. There are four possible settings each with slightly different behavior.

No Preference Omits the compression directives from the OpenVPN configuration entirely. No compression will be performed, but this may be overridden by other methods such as Client-Specific overrides or advanced options.

Disabled - No Compression Explicitly disables compression in the configuration

Enabled with Adaptive Compression Enables compression with a periodic test to ensure the traffic is able to be compressed. If compression is not optimal, it will be disabled until it is tested again. This option strikes the best balance since it will compress data when it will help, but does not compress data when it is hindering performance.

Enabled without Adaptive Compression Explicitly enables compression to be on at all times without testing the traffic.

Type-of-Service

When this option is enabled OpenVPN will set the Type-of-Service (TOS) IP header value of tunnel packets to match the encapsulated packet value. This may cause some important traffic to be handled faster over the tunnel by intermediate hops, at the cost of some minor information disclosure.

The most common example is VoIP or video traffic. If the TOS bit is set to reflect the priority of the traffic it can help QoS along the path, but someone intercepting the traffic could see the TOS bit and gain some knowledge about the contents of the traffic inside the tunnel. For those who rely on TOS bits for QoS, the benefit may outweigh the information leak.

Inter-Client Communication

This option controls whether or not connected clients are able to communicate with one another. To allow this behavior, check the option. When unchecked, clients can only send traffic to the server or destinations beyond the server such as routed networks or the Internet.

Typically in remote access style deployments it is unnecessary for clients to reach each other, but there are some corner cases when it can be helpful. One example is remote web developers working together and running test servers on their local systems. With this option activated, they can reach the other test serves for collaborative development.

Duplicate Connections

By default OpenVPN will associate an IP address from its tunnel network with a specific certificate or username for a given session. If the same certificate connects again, it would be assigned the same IP address and either disconnect the first client or cause an IP conflict where neither client will receive proper data. This is primarily for security reasons so the same certificate cannot be used by multiple people simultaneously. We recommend a unique certificate be used for each connecting user. Otherwise if a client is compromised there is no way to revoke that one client alone, certificates would need to be reissued to all clients that share the same certificate.

If a setup that uses the same certificate in multiple locations is an absolute requirement and cannot be avoided, check **Duplicate Connections** to allow the non-standard behavior of multiple clients using the same certificate or username.

Disable IPv6

When checked, IPv6 traffic forwarding is disabled for this OpenVPN instance.

Client Settings

These settings pertain to how clients connecting to this sever instance will behave.

Dynamic IP

Checking this box adds the *float* configuration option to the OpenVPN configuration. This allows clients to retain their connection if their IP address changes. similar to MOBIKE for IKEv2 in IPsec. For clients on Internet connections where the IP changes frequently, or mobile users who commonly move between different Internet connections, check this option to allow for stable connectivity. Where the client IP is static or rarely changes, not using this option offers a small security improvement.

Address Pool

When this option is enabled the server will assign virtual adapter IP addresses to clients from the subnet specified by the **Tunnel Network** option. When unchecked IP addresses will not be assigned automatically and clients will have to set their own static IP addresses manually in their client configuration files. Except in rare cases, this is almost always enabled.

Topology

By default OpenVPN on pfSense® software version 2.3 and later prefers a topology style of *subnet* when using a **Device Mode** of *tun*. This style allocates only one IP address per client rather than an isolated subnet per client. This is the only available style when using the *tap* **Device Mode**.

When the older *net30* topology for *tun* is chosen, OpenVPN allocates a /30 CIDR network (four IP addresses, two usable) to each connecting client. This style has a longer history, but can be confusing for administrators and users alike.

The **Topology** option is relevant only when supplying a virtual adapter IP address to clients using *tun* mode on IPv4. Some clients may require this even for IPv6, such as OpenVPN Connect, though in reality IPv6 always runs with a subnet topology even when IPv4 uses *net30*. OpenVPN version 2.1.3 or newer is required to use a *subnet* topology, and there were significant fixes to it in OpenVPN 2.3 as well, so using a current OpenVPN client version is important.

> **Warning:** The default in pfSense has been changed to *subnet* because the OpenVPN project has declared the *net30* style as deprecated, indicating it will be removed in future versions.
>
> Be aware, however, that some very old clients may break if this option is used, such as older versions of OpenVPN (Before 2.0.9, released nearly 10 years ago), Windows versions with older tun/tap drivers, or clients such as Yealink phones. Always make sure the client and associated drivers are fully up-to-date when using a *subnet* topology.

DNS Default Domain

When checked, a field will appear to specify the DNS domain name to be assigned to clients. To ensure name resolution works properly for hosts on the local network where DNS name resolution is used, specify the internal DNS domain name here. For Microsoft Active Directory environments, this would usually be the Active Directory domain name.

DNS servers

When checked, up to four DNS servers may be entered for use by the client while connected to the VPN. For Microsoft Active Directory environments, this is typically the Active Directory Domain Controllers or DNS servers for proper name resolution and authentication when connected via OpenVPN.

Force DNS Cache Update

When checked, this option will push a set of commands to Windows clients that will flush their DNS and restart caching to improve client handling of updated DNS servers from the VPN.

NTP servers

When checked, one or two NTP servers may be set for syncing clocks on clients. It can be an IP address or FQDN.

NetBIOS Options

When **Enable NetBIOS over TCP/IP** is checked, several other NetBIOS and WINS related options will appear. If the box is unchecked, these settings will be disabled.

Node Type

The NetBIOS node type controls how Windows systems will function when resolving NetBIOS names. It's usually fine to leave this to *none* to accept Windows' default.

The available options include:

b-node Use broadcasts for NetBIOS name resolution. This would not be used except in the case of a tap bridge.

p-node Point-to-point name queries to a WINS server. WINS has been mostly deprecated, so this option is not useful in modern Windows networks.

m-node Broadcast then query name server. Similar to b-node but will fall back to DNS.

h-node Query name server first, then use broadcast. This option is the most likely to succeed in a current network with proper, functional, DNS.

Scope ID

A NetBIOS Scope ID provides an extended naming service for NetBIOS over TCP/IP. The NetBIOS scope ID isolates NetBIOS traffic on a single network to only those nodes with the same NetBIOS scope ID.

WINS Servers

Checking this box allows two WINS servers to be defined which provides name resolution for clients accessing and browsing NetBIOS resources across the VPN. WINS has been largely deprecated and removed from use, so it's unlikely this will be needed in most modern environments.

Enable Custom Port

When checked, a non-default **Management Port** may be specified for use with the OpenVPNManage feature of the OpenVPN Client Export package. If multiple connections profiles are used on a single client using that interface, each must use a unique management port.

Custom options

While the pfSense web interface supports the most commonly used options, OpenVPN is very powerful and flexible and occasionally options that are unavailable in the web interface may be necessary. Such custom options may be added in using this entry box. These options are described further in *Custom configuration options*.

Verbosity level

Configures the amount of detail shown in the OpenVPN logs for this instance, useful for troubleshooting problems. Higher numbers will result in higher amounts of detail in the log. During normal operation the *default* selection is best.

Note: When set to higher levels, the OpenVPN status page and dashboard widget will cause additional logging as they interact with the Management process to poll information from the OpenVPN daemons.

Client Configuration Options

These options are available in one or more modes for OpenVPN client instances, managed from **VPN > OpenVPN**, on the **Clients** tab.

Many of these options are identical to the server options mentioned above, so only differences will be noted.

Server mode

For client instances, the server mode choices are limited to *Peer to Peer (SSL/TLS)* and *Peer to Peer (Shared Key)*, which pair with the server options of the same name and type.

Interface

This option selects the interface, VIP, or failover group that the OpenVPN client instance will use for outgoing connections.

When a CARP type VIP is selected for the Interface on OpenVPN Client instances, the OpenVPN instance will be stopped when the CARP VIP is in a backup state. This is done to prevent the secondary HA node from maintaining invalid routes or attempting to make outbound connections which can interfere with the active connection on the primary HA node.

Local Port

For clients, the local port is left blank in nearly every case so that a randomized local port will be used. This is more secure, but some implementations may require a specific source port. If a specific source port is required, fill it in as needed as needed.

Server host or address

The IP address or fully qualified domain name for the server.

Note: When using a hostname for the remote server address, the server host name will be resolved on each connection attempt.

Server Port

The port on which the server is listening, typically `1194`

Proxy Settings

Proxy Host or Address The IP address or fully qualified domain name for a proxy server through which this client must connect.

Proxy Auth Extra Options Extra authentication options. When set to *basic* or *ntlm*, **Username** and **Password** fields are presented so that proxy authentication may be configured.

User Authentication Settings

When using *Peer to Peer SSL/TLS* mode, a **Username** and **Password** may be specified in addition to, or instead of, a user certificate, depending on the requirements configured on the server.

Cryptographic Settings

The settings in this section are identical to those on their corresponding options on the server side except for the new **Client Certificate** option, where the certificate is selected for use by this client. This certificate (and the associated key, and CA Certificate) must be imported to this firewall before they can be chosen.

Shared Key / TLS Authentication

These options work similar to the server side counterparts, but be aware that the key from the server must be copied here, rather than generating a new key on the client.

Limit Outgoing Bandwidth

The value in this box, specified in **bytes per second**, is used to limit the speed of outgoing VPN traffic. When left blank, there is no limit. The value must be between *100* and *100000000*.

Don't Pull Routes

When checked, the client will ignore routes pushed from the server. This is useful in cases when the server pushes a default gateway redirect when this client does not need one.

Don't Add/Remove Routes

When checked, OpenVPN will not manage route table entries for this VPN. In this case, they must be managed manually. The routes that would normally be added are instead passed to `--route-upscript` using environmental variables.

19.7.3 Custom configuration options

OpenVPN offers dozens of configuration options, many beyond the most commonly used fields presented in the GUI. This is why the **Advanced configuration** box exists. Additional configuration options may be configured using this input area, separated by semicolons.

This section covers the most frequently used custom options individually. There are many more, though rarely needed. The OpenVPN man page details them all.

Warning: Exercise caution when adding custom options, there is no input validation applied to ensure the validity of options used. If an option is used incorrectly, the OpenVPN client or server may not start. View the OpenVPN logs under **Status > System logs** on the **OpenVPN** tab to ensure the options used are valid. Any invalid options will result in a log message, followed by the option that caused the error:

```
Options error: Unrecognized option or missing parameter(s)
```

Routing options

To add additional routes for a particular OpenVPN client or server, use the **Local Network** and **Remote Network** boxes as needed, using a comma-separated list of networks.

The `route` custom configuration option may also be used, but is no longer necessary. Some users prefer this method, however. The following example adds a route for `10.50.0.0/24`:

```
route 10.50.0.0 255.255.255.0;
```

To add multiple routes, separate them with a semicolon:

```
route 10.50.0.0 255.255.255.0;
route 10.254.0.0 255.255.255.0;
```

The `route` configuration option is used to add routes locally for networks that are reachable through the VPN. For an OpenVPN server configuration using PKI, additional routes may also be pushed to clients. The GUI can configure these using the **Local Network** field. To push the routes manually for `10.50.0.0/24` and `10.254.0.0/24` to all clients, use the following custom configuration option:

```
push "route 10.50.0.0 255.255.255.0";
push "route 10.254.0.0 255.255.255.0";
```

Redirecting the default gateway

OpenVPN also allows the default gateway to be redirected across the VPN, so all non-local traffic from the client is sent through the VPN. This is great for untrusted local networks such as wireless hotspots, as it provides protection against numerous attacks that are a risk on untrusted networks. This is configurable in the GUI now, using the **Redirect Gateway** checkbox in the OpenVPN instance configuration. To do this manually, add the following custom option:

```
push "redirect-gateway def1"
```

The same value may be used as a custom option on the client side by entering `redirect-gateway def1` without specifying `push` . (Note the option is the letters "def" followed by the digit *one*, not the letter "L".)

19.7.4 OpenVPN Firewall Rules

Permitting traffic to the OpenVPN server

A firewall rule must permit traffic to the OpenVPN server or clients will not be able to connect. Add a rule as follows:

- Navigate to **Firewall > Rules**, **WAN** tab
- Click  to create a new rule at the top of the list
- Set **Protocol** to *UDP*
- Leave the **Source** set to *any*
- Set the **Destination** to *WAN Address*
- Set the **Destination port** to `1194` in this instance (or whichever port the server is using to listen)
- Enter a **Description**, such as `Allow traffic to OpenVPN Server`
- Click **Save**
- Click **Apply changes**

This rule is depicted in Figure *OpenVPN Server WAN Rule*.

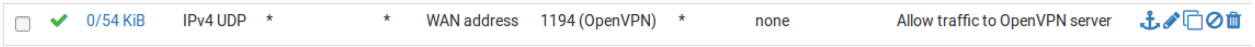


Fig. 1: OpenVPN Server WAN Rule

If the client source addresses are known and do not change, then the source of the rule could be altered to limit traffic from only those clients. This is more secure than leaving the server exposed to the entire Internet, but that is necessary to accommodate clients with dynamic IP addresses, roaming clients, and so on. The risk of leaving the service exposed with most OpenVPN configurations is minimal, especially in cases where TLS Authentication is employed. With certificate based authentication there is less risk of compromise than password-based solutions that are susceptible to brute forcing. This presumes a lack of security holes in OpenVPN itself, which to date has a solid security track record.

Allowing traffic over OpenVPN Tunnels

By default, all traffic is blocked from entering OpenVPN tunnels. To allow traffic from remote OpenVPN nodes to make connections to resources on the local side, firewall rules under **Firewall > Rules**, on the **OpenVPN** tab are required.

As with other aspects of the firewall, these rules will only match traffic coming into the firewall from the remote side, not traffic leaving from this side, so craft the rules accordingly. In cases when pfSense® software is on both ends and traffic is required to reach between local networks on both sides, then rules are required on both firewalls.

Add an OpenVPN rule which passes all traffic as follows:

- Navigate to **Firewall > Rules**, **OpenVPN** tab
- Click  to create a new rule at the top of the list
- Set **Protocol** to *any*
- Enter a **Description** such as `Allow all on OpenVPN`
- Click **Save**
- Click **Apply changes**

To limit the traffic to only specific sources and destinations, adjust the rule(s) as needed. A strict ruleset is more secure, but more difficult to create.

Tip: Rules on the OpenVPN tab apply to all OpenVPN server and client instances. The OpenVPN interface may also be assigned (*Assigning OpenVPN Interfaces*) in which case there will be a separate firewall rule tab for that VPN, upon which rules can pass traffic for that specific VPN.

19.7.5 OpenVPN clients and Internet Access

For OpenVPN Remote Access clients to reach the Internet through the OpenVPN connection, Outbound NAT is required to translate their traffic to the WAN IP address of the firewall. The default Automatic Outbound NAT rules cover this, but if Manual Outbound NAT is in use, manual rules are necessary to perform outbound NAT on traffic from sources that include the OpenVPN tunnel network or remote network(s).

See also:

Outbound NAT for more details on Outbound NAT.

19.7.6 Assigning OpenVPN Interfaces

In order to do complex NAT, policy routing, or tunnel-specific filtering, the OpenVPN interface must be assigned as an OPT interface and configured accordingly.

Assigning the OpenVPN interface enables several beneficial changes for advanced control of VPN traffic:

- Adds a firewall tab under **Firewall > Rules**
- Adds *reply-to* to rules on the VPN interface tab to help with return routing
- Adds a Gateway entry for the far side of the VPN for policy routing
- Allows the interface to be selected elsewhere in the GUI and packages
- Allows more fine-grained control of Port Forwards and Outbound NAT for the VPN

Interface assignment and configuration

- Navigate to **Interfaces > Assignments**
- Select the appropriate `ovpns` or `ovpnc` interface in **Available network ports**, the description of the VPN is printed for reference.
- Click  **Add** to assign the interface as a new OPT interface (e.g. OPT1)

Figure *Assign OpenVPN Interface* shows `ovpns1` assigned as OPT1.

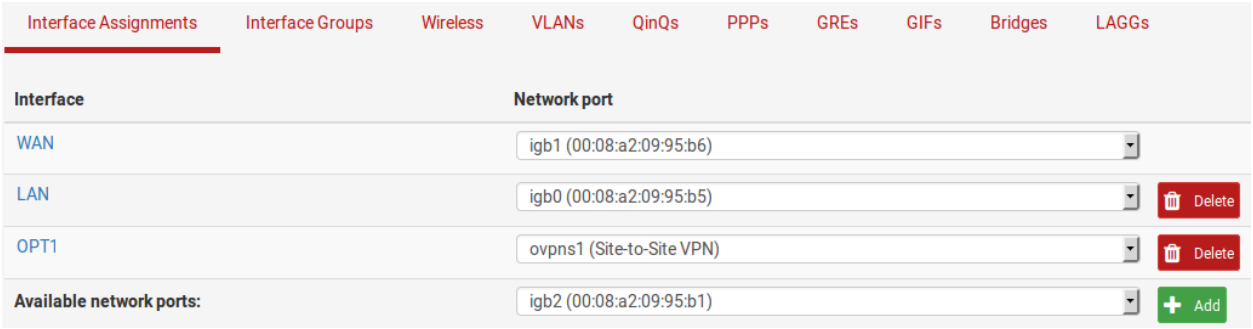


Fig. 2: Assign OpenVPN Interface

- Navigate to the Interface configuration page, **Interfaces > OPTx**
- Check **Enable**
- Enter an appropriate **Description** which will become the interface name (e.g. `VPNServer`)
- Select *none* for both **IPv4 Configuration Type** and for **IPv6 Configuration Type**

> **Note:** This will not configure any IP address information on the interface, which is necessary since OpenVPN itself must configure these settings.

- Click **Save**
- Click **Apply Changes**

This does not change the functionality of OpenVPN, it makes the interface available for firewall rule, NAT, and gateway purposes, among other uses.

After assigning the OpenVPN interface, edit the OpenVPN server or client and click **Save** once there as well to reinitialize the VPN. This is necessary for the VPN to recover from the assignment process.

Filtering with OpenVPN

When the OpenVPN interface is assigned, a tab is present under **Firewall > Rules** dedicated to only this single VPN. These rules govern traffic coming in from the remote side of the VPN and they even get the pf `reply-to` keyword which ensures traffic entering this VPN interface will exit back out the same interface. This can help with some more advanced NAT and configuration scenarios.

Note: Rules added here are processed *after* the OpenVPN tab rules, which are checked first. In order to match the rules on an assigned VPN tab, the traffic **must not match** any rules on the OpenVPN tab. Remove any "Allow All"

style rules from the OpenVPN tab and craft more specific rules instead.

See also:

For more information on firewall rules, refer to *Firewall*.

Policy Routing with OpenVPN

When the OpenVPN interface is assigned and enabled, an automatic gateway entry is added under **System > Routing**, on the **Gateways** tab. With this, traffic can be directed into the VPN using the **Gateway** field on LAN or other internal interface firewall rules.

When used with a VPN to reach Internet sites, more configuration may be required. Either outbound NAT must be performed on the VPN interface before it leaves (for VPN services such as PIA, StrongVPN and similar) or the NAT must be done on the other side before it reaches the actual Internet connection.

See also:

See *Policy routing* for more information on policy routing.

Warning: Do not use this automatic gateway for static routes. Use the **Remote Network** field in the VPN configuration. Defining a static route using the automatic OpenVPN gateway will not work properly.

NAT with OpenVPN

When the OpenVPN interface is assigned NAT rules can also be applied the same as with any other interface. This is useful when connecting two conflicting subnets or for making NAT rules specific to this one VPN connection (outbound NAT, port forwards, or 1:1 NAT)

19.7.7 OpenVPN and Multi-WAN

OpenVPN is multi-WAN capable, with some caveats in certain circumstances. This section covers multi-WAN considerations with OpenVPN server and client configurations.

OpenVPN assigned to a Gateway Group

A Gateway Group (*Gateway Groups*) may be selected as the **Interface** for an OpenVPN instance. Such a gateway group must be configured for failover only, not load balancing. Failover groups only have one gateway per tier. When creating the gateway group, a VIP may also be chosen for use with a specific gateway. When selected for a VPN server, the interface or VIP of the Tier 1 gateway in the group will be used first. If that gateway goes down, it will move to tier 2, and so on. If the tier 1 gateway comes back up, the VPN will resume operating on that WAN immediately. When used for a VPN server, this means that the server is only active on one WAN at a time. Some of the other methods described below may be better for most common circumstances, such as needing both WANs usable concurrently with the VPN. When used with OpenVPN clients, the outbound interface will be switched according to the gateway group tiers.

OpenVPN servers and multi-WAN

OpenVPN servers can be used with any WAN connection, though the means of doing so will vary depending on the specifics of a given configuration.

OpenVPN server using TCP

TCP is not the preferred protocol for OpenVPN. However, using TCP can make multi-WAN OpenVPN easier to configure when the VPN is using an interface setting of *any*. OpenVPN servers using TCP will work properly on all WANs where the firewall rules allow the traffic to the OpenVPN server. A firewall rule is required for each WAN interface. This method should be considered a last resort, and only used if the other methods are not viable.

Note: This works because of the connection-oriented nature of TCP. The OpenVPN can reply back to the other end with the proper source preserved since it is part of an open connection.

OpenVPN server using UDP

OpenVPN servers with UDP are also multi-WAN capable, but with some caveats that aren't applicable with TCP.

These OpenVPN limitations are due to the connectionless nature of UDP. The OpenVPN instance replies back to the client, but the Operating System selects the route and source address based on what the routing table believes is the best path to reach the other side. For non-default WANs, that will not be the correct path.

Multiple Server Method

In some cases, each WAN must have its own OpenVPN server. The same certificates may be for all the servers. Only two parts of the OpenVPN configuration must change:

Tunnel Network Each server must have a unique **Tunnel Network** that does not overlap with any other tunnel network or internal subnet.

Interface Each OpenVPN server must specify a different WAN **Interface**.

Port forward method

An easier and more flexible option is to bind the OpenVPN server to the *LAN* interface or *Localhost* and use a port forward from each WAN to direct the OpenVPN port to the service. Using this method the reply-to functionality in pf will ensure that the return traffic flows back to the proper source via the intended interface.

This method requires some minor manual intervention when used with the client export package. The **Host Name Resolution** option must be set to one of the automatic port forward methods otherwise the default export settings would leave it attempting to connect to the wrong address. See *OpenVPN Client Export Package* for details

Automatic Failover for Clients

Multiple remote servers can be configured on OpenVPN clients. If the first server cannot be reached, the second will be used. This can be used in combination with a multi-WAN OpenVPN server deployment to provide automatic failover for clients. If the OpenVPN servers are running on IP addresses 198.51.100.3 and 203.0.113.5, both using port 1194, the `remote` lines in the client configuration file will be as follows:

```
remote 198.51.100.3 1194 udp
remote 203.0.113.5 1194 udp
```

For clients configured on pfSense® software, the first `remote` is configured by the **Server Host or Address* field in the GUI. The second ``remote`` is specified in the **Advanced** field.

This method has three notable behaviors that some may find undesirable:

- It will take at least 60 seconds to detect a failure and switch to the next server.
- Any connection failure will cause it to try the second server, even if it is not a WAN failure.
- It will **not** "fail-back". Once a client connects to the second server IP address it will stay there until disconnected.

OpenVPN Clients and Multi-WAN

To use an OPT WAN interface, select it as the **Interface**. OpenVPN clients configured on the firewall will respect the chosen **Interface** and a static route is added automatically behind the scenes to ensure traffic takes the correct path.

If the interface is instead set to *any*, the client will follow the system routing table when making the connection to the OpenVPN server. In this case a manual static route will be required to direct traffic to the remote endpoint over the desired WAN.

OpenVPN Site-to-Site with Multi-WAN and OSPF

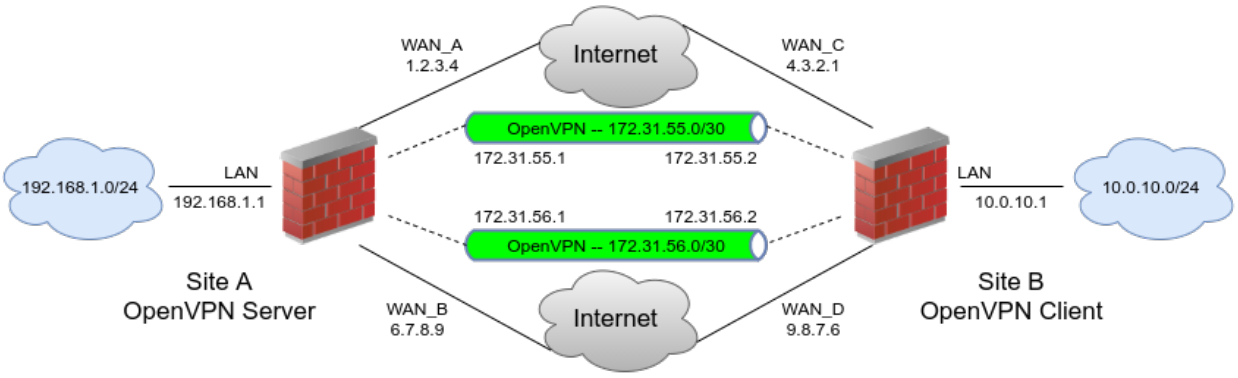


Fig. 3: Example OpenVPN Setup Involving OSPF Across Multiple WANs

Building upon concepts from earlier in the chapter, it is possible to configure a redundant VPN using a dynamic routing protocol such as OSPF as seen in Figure *Example OpenVPN Setup Involving OSPF Across Multiple WANs*.

First, setup **shared key** site-to-site OpenVPN instances on each WAN for the remote sites. **Do not fill in** the **Remote Networks** fields on either side, only **Tunnel Network** addresses.

- Setup two servers on the local side, each on a different port. Use two distinct, non-overlapping tunnel networks (e.g. `172.31.55.0/30` and `172.31.56.0/30`)

- Setup two clients on the remote firewall, each paired up with one of the above servers, matching the IP addresses and port numbers involved.
- Ensure the clients are set for their specific WAN, choose the interface from the drop-down menu, or a CARP VIP that is on one of the WANs being used.
- Ensure these OpenVPN connections link up between client and server. The tunnel address on both sides will respond to a ping when they are working correctly. If the tunnels do not establish, see *Troubleshooting OpenVPN* for suggestions on troubleshooting the connection.
- Ensure the OpenVPN firewall rules allow all traffic or at least allow OSPF traffic from a source of the tunnel networks to a destination of any. The destination on the traffic will be a multicast address, which can be used to filter specifically if needed, but there isn't much to be gained in the way of security if the source is locked down in the rules as the traffic cannot leave that segment.

Once both instances are connected, configure OSPF.

- Install the FRR package from **System > Packages**, **Available Packages** tab **on both firewalls**.
- Navigate to **Services > FRR OSPF**, **Interfaces tab**
- Add each OpenVPN interface
  - Set the cost to `10` on the primary link and `20` on the secondary, and so on
  - Add the LAN and other internal interfaces as **passive** interfaces
- Navigate to the **[Global Settings]** tab
- Enter a Master Password. It doesn't matter what it's set to, it is used internally for accessing the status daemon.
- Set the Router ID to an IP-address-like value, (e.g. `10.3.0.1`.) The Router ID is unique on each device, which is why setting it to the LAN IP address of a router is a good practice.
- Set the Area ID which is also an IP-address-like value. The Area ID is typically set to `0.0.0.0` or `0.0.0.1`, but any properly-formatted value may be used. The Area ID is the **same** for **all** routers involved in this VPN
- Click **Save**

See also:

- *Open Shortest Path First v2 (OSPF)*

Once OSPF has been configured on all routers, they will attempt to form a neighbor relationship.

After OSPF has been setup on both ends the **Status** tab will show a full peering with each instance on each wan if they connected properly, and the routes obtained via OSPF will be listed. Once that happens, try unplugging/replugging WANs and refreshing the status while running some test traffic across the VPN, such as an ICMP ping.

19.7.8 OpenVPN and CARP

OpenVPN works well with High Availability using CARP. To provide a high availability OpenVPN solution with CARP, configure the OpenVPN server or client to use the CARP VIP with the **Interface** option and configure clients to connect to that CARP VIP.

When XMLRPC Configuration Synchronization settings are enabled, OpenVPN instances will automatically synchronize. The connection state isn't retained between hosts so clients must reconnect after failover occurs, but OpenVPN will detect the connection failure and reconnect within a minute or so of failover. High Availability and CARP are discussed further in *High Availability*.

When a CARP VIP is selected as the **Interface** for an OpenVPN instance the firewall will automatically shut down OpenVPN client instances as needed when a CARP node is in a BACKUP state. This prevents OpenVPN from making

unnecessary outbound connections in client mode. When the CARP VIP status transitions to MASTER, the OpenVPN instances are started automatically.

19.7.9 Sharing a Port with OpenVPN and a Web Server

To be extra sneaky or careful with an OpenVPN server, take advantage of the `port-share` capability in OpenVPN that allows it to pass any non-OpenVPN traffic to another IP address behind the firewall. The usual use case for this would be to run the OpenVPN server on port tcp/443 while letting OpenVPN hand off the HTTPS traffic to a web server in place of a port forward.

Often on locked-down networks, only ports like 80 and 443 will be allowed out for security reasons and running OpenVPN instances on these allowed ports can help users get out in situations where access may otherwise be restricted.

To set this up, configure an OpenVPN server to listen on TCP port 443 and add a firewall rule to pass traffic to the WAN IP address or VIP used for OpenVPN on port 443. No additional port forwards or firewall rules are necessary to pass the traffic to the internal IP.

In the custom options of the OpenVPN instance, add the following:

```
port-share x.x.x.x 443
```

Where `x.x.x.x` is the internal IP address of the web server to which the non-VPN traffic will be forwarded by OpenVPN.

Now if an OpenVPN client is pointed to the public address it will connect to the VPN, and if a web browser is pointed at the same IP address, it will be connected to the web server.

Note: This requires using TCP, and will likely result in reduced VPN performance.

19.7.10 Controlling Client Parameters via RADIUS

When using RADIUS as an authentication source for a VPN, pfSense® software supports receiving some client configuration parameters from the RADIUS server as reply attributes. The following values may be specified:

Cisco-AVPair = <IP_PROTO>:inacl#<NUM>= Inbound firewall rules to govern traffic from the client to the server, where `<IP_PROTO>` is IP protocol (`ip` or `ipv6`) and `<NUM>` is a rule number. Given in Cisco-style ACL format subnet masks are specified wildcard style. It is possible to use client variables `{clientip}` and `{clientipv6}` which are replaced with the connecting client Tunnel IP addresses. FreeRADIUS example:

```
Cisco-AVPair = "ip:inacl#1=permit tcp host 192.168.5.10 host 192.168.6.3␣
↪eq 80",
Cisco-AVPair += "ip:inacl#2=permit udp host {clientip} host 192.168.33.4␣
↪eq 53",
Cisco-AVPair += "ipv6:inacl#1=permit icmp host {clientipv6} host␣
↪2001:DB8::10",
Cisco-AVPair += "ipv6:inacl#2=permit udp host 2001:DB8::4444 host␣
↪2001:DB8::7 range 1024 65535"
```

Cisco-AVPair = <IP_PROTO>:outacl#<NUM>= Outbound firewall rules to govern traffic from the server to the client. Formatted the same as the `inacl` parameter.

Cisco-AVPair dns-servers= DNS servers to push to the client. Multiple servers may be specified, separated by spaces.

Cisco-AVPair route= Additional route statements to push to the client. Specified as `x.x.x.x y.y.y.y` where the first parameter is a network address and the second is a subnet mask.

Framed-IP-Address= The IP address to assign to the client. When using a *subnet* style **Topology** the RADIUS server must also send back a Framed-Mask set appropriately for the **Tunnel Network** of the VPN. When using a *net30* style **Topology**, the client receives this IP address and the server side is set as one IP address lower than the address given to the client.

19.7.11 OpenVPN Adapter Address ICMP Behavior

Sometimes OpenVPN will not respond to ping on certain virtual addresses used solely for routing endpoints when using the *net30* topology. Do not rely on pinging the OpenVPN endpoint addresses as a means of determining if the tunnel is passing traffic properly. Instead, ping something in the remote subnet, such as the LAN IP of the server.

According to the OpenVPN FAQ, in the section titled *Why does OpenVPN's "ifconfig-pool" option use a /30 subnet (4 private IP addresses per client) when used in TUN mode?*:

> As 192.168.1.5 is only a virtual IP address inside the OpenVPN server, used as an endpoint for routes, OpenVPN doesn't bother to answer pings on this address, while the 192.168.1.1 is a real IP address in the servers O/S, so it will reply to pings.

This may seem a little counter-intuitive, since on the server something like this is seen in the ifconfig output:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
        inet6 fe80::202:b3ff:fe03:8028%tun0 prefixlen 64 scopeid 0xc
        inet 192.168.100.1 --> 192.168.100.2 netmask 0xffffffff
        Opened by PID 27841
```

While the client shows:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
        inet6 fe80::202:b3ff:fe24:978c%tun0 prefixlen 64 scopeid 0xa
        inet 192.168.100.6 --> 192.168.100.5 netmask 0xffffffff
        Opened by PID 1949
```

In this case, *.5* or *.1* will not likely respond to ping. *.5* because it's a virtual address, and *.1* because the client has no route to reach it directly. The *.5* and *.6* addresses are part of a */30* that goes from *.4* to *.7*, and trying to ping *.1* would go out the default route instead.

There are many cases where the far side of an OpenVPN tunnel **can** respond to ping, but not the local. This is also counter-intuitive, but works especially in cases where there is a site-to-site link. If the server shows its tun addresses as "x.x.x.1 -> x.x.x.2" and the client shows the reverse - "x.x.x.2 -> x.x.x.1", then the far side will likely respond to ping from both ends.

In contrast, when using "topology subnet" these virtual addresses and /30 subnets are not used so these issues are not present.

See also:

- *OpenVPN Client Export Package*
- *Checking the Status of OpenVPN Clients and Servers*
- *OpenVPN Logs*
- *Connecting OpenVPN Sites with Conflicting IP Subnets*
- *OpenVPN Remote Access Configuration Example*
- *Authenticating OpenVPN Users with FreeRADIUS*

- *Authenticating OpenVPN Users with RADIUS via Active Directory*
- *Installing OpenVPN Remote Access Clients*
- *Installing the OpenVPN Client on iOS*
- *Adding OpenVPN Remote Access Users*
- *OpenVPN Site-to-Site Configuration Example with Shared Key*
- *Routing Internet Traffic Through A Site-To-Site OpenVPN Tunnel*
- *OpenVPN Site-to-Site Configuration Example with SSL/TLS*
- *Configuring a Single Multi-Purpose OpenVPN Instance*
- *Bridging OpenVPN Connections to Local Networks*
- *Connecting to an OpenVPN Access Server*
- *Troubleshooting OpenVPN*
- *Troubleshooting OpenVPN Internal Routing (iroute)*
- *Troubleshooting OpenVPN Push Routes*
- *Troubleshooting OpenVPN Remote Access Client IP Address Assignments*
- *Troubleshooting Windows OpenVPN Client Connectivity*
- *Troubleshooting Windows/SMB Share Access from OpenVPN Clients*

OpenVPN is an open source SSL VPN solution that can be used for remote access clients and site-to-site connectivity. OpenVPN supports clients on a wide range of operating systems including all the BSDs, Linux, Android, Mac OS X, iOS, Solaris, Windows 2000 and newer, and even some VoIP handsets.

Every OpenVPN connection, whether remote access or site-to-site, consists of a server and a client. In the case of site-to-site VPNs, one firewall acts as the server and the other as the client. It does not matter which firewall possesses these roles. Typically the location of the primary firewall will provide server connectivity for all remote locations, whose firewalls are configured as clients. This is functionally equivalent to the opposite configuration the primary location configured as a client connecting to servers running on the firewalls at the remote locations. In practice, the servers are nearly always run on a central location.

There are several types of authentication methods that can be used with OpenVPN: shared key, X.509 (also known as SSL/TLS or PKI), user authentication via local, LDAP, and RADIUS, or a combination of X.509 and user authentication. For shared key, a single key is generated that will be used on both sides. SSL/TLS involves using a trusted set of certificates and keys. User authentication can be configured with or without SSL/TLS, but its use is preferable where possible due to the increased security is offers.

The settings for an OpenVPN instance are covered in this chapter as well as a run-through of the OpenVPN Remote Access Server wizard, client configurations, and examples of multiple site-to-site connection scenarios.

Note: While OpenVPN is an SSL VPN, it is not a "clientless" SSL VPN in the sense that commercial firewall vendors commonly state. The OpenVPN client must be installed on all client devices. In reality no VPN solution is truly "clientless", and this terminology is nothing more than a marketing ploy. For more in depth discussion on SSL VPNs, this post from Matthew Grooms, an IPsec tools and pfSense® developer, in the mailing list archives provides some excellent information.

For general discussion of the various types of VPNs available in pfSense and their pros and cons, see *Virtual Private Networks*.

See also:

pfSense Hangouts on Youtube to view the September 2014 Hangout on Advanced OpenVPN Concepts and the September 2015 Hangout on Remote Access VPNs

19.7.12 OpenVPN and Certificates

Using certificates is the preferred means of running remote access VPNs, because it allows access to be revoked for individual machines. With shared keys, either a unique server and port for must be created for each client, or the same key must be distributed to all clients. The former gets to be a management nightmare, and the latter is problematic in the case of a compromised key. If a client machine is compromised, stolen, or lost, or otherwise needs revoked, the shared key must be re-issued to all clients. With a PKI deployment, if a client is compromised, or access needs to be revoked for any other reason, simply revoke that client's certificate. No other clients are affected.

The pfSense GUI includes a certificate management interface that is fully integrated with OpenVPN. Certificate authorities (CAs) and server certificates are managed in the Certificate Manager in the web interface, located at **System > Cert Manager**. User certificates are also managed in the web interface, as a part of the built-in user manager found at **System > User Manager**. Certificates may be generated for any user account created locally on the firewall except for the default admin account. For further information on creating a certificate authority, certificates, and certificate revocation lists, see *Certificate Management*.

19.8 IPsec

19.8.1 IPsec Configuration

IPsec offers numerous configuration options, affecting the performance and security of IPsec connections. Realistically, for low to moderate bandwidth usage it matters little which options are chosen here as long as DES is not used, and a strong pre-shared key is defined, unless the traffic being protected is so valuable that an adversary with many millions of dollars worth of processing power is willing to devote it to breaking the IPsec encryption. Even in that case, there is likely an easier and much cheaper way to break into the network and achieve the same end result (social engineering, for one). Performance is the most important factor for most, and in cases when that is a concern, more care is needed when crafting a configuration.

- *IPsec Modes*
- *Interface Selection*
- *Phase 1 Settings*
- *Phase 2 Settings*

IPsec Modes

pfSense® software supports several primary modes of IPsec operation:

Policy-based IPsec This mode uses policies to match specific combinations of traffic which are grabbed by the kernel and pushed through an IPsec tunnel. It also uses special "trap" policies to detect when traffic intends to use IPsec so that it can bring the tunnel up automatically. Only traffic specifically matching phase 2 child SA entries can use IPsec, and all traffic matching those entries will be taken over by IPsec.

This mode is the most common and is supported by nearly all third party IPsec implementations.

Route-based IPsec (VTI) Routed IPsec uses a special Virtual Tunnel Interface (VTI) for each IPsec tunnel. The VTI interface is assigned and used like other interfaces. Phase 2 entries define addresses for the tunnel interface itself, rather than policies which direct traffic to IPsec. Arbitrary traffic may cross IPsec tunnels, as traffic follows the system routing table. Static routes or dynamic routing daemons can control which traffic crosses a tunnel.

Support for routed IPsec varies by vendor.

Mobile IPsec Similar to policy-based mode, but for remote access/mobile clients.

Transport Mode This mode encrypts all traffic from the the external IP address on this firewall to the external IP address on the far side as defined in the Phase 1 settings. Since all traffic sent between the two nodes will be encrypted, other tunneling methods that do not employ encryption, such as a GIF or GRE tunnel, can be safely used by the firewall between the endpoints.

Interface Selection

In many cases, the **Interface** option for an IPsec tunnel will be WAN, since the tunnels are connecting to remote sites. However, there are plenty of exceptions, the most common of which are outlined in the remainder of this section.

CARP Environments

CARP type virtual IP addresses are also available in the **Interface** drop-down menu for use in High Availability environments (*High Availability*). In these environments, an appropriate CARP address must be chosen for the WAN where the IPsec tunnel will terminate. By using the CARP IP address, it ensures that the IPsec tunnel will be handled by the High Availability cluster member currently in MASTER state, so even if the primary firewall is down, the tunnel will connect to whichever cluster member has taken over the MASTER role.

IP Alias VIP

If multiple IP addresses are available on an interface using IP Alias type VIPs, they will also be available in this list. To use one of those IP addresses for the VPN instead, select it here.

Multi-WAN Environments

When using Multi-WAN (*Multiple WAN Connections*), pick the appropriate Interface choice for the WAN-type interface to which the tunnel will connect. If the connection will enter via WAN, pick WAN. If the tunnel will use a different WAN, choose whichever OPT WAN interface is needed. A static route will automatically be added to ensure that the traffic to the **Remote Gateway** routes through the appropriate WAN.

A gateway group may also be chosen from this list. A gateway group to be used with IPsec must only have *one* gateway per tier. When using a gateway group, if the first gateway goes down, the tunnel will move to the next available WAN in the group. When the first WAN comes back up, the tunnel will be rebuilt there again. If the endpoint on the far side is one that does not support multiple peer addresses, such as another firewall running pfSense software, this must be combined with a DynDNS host set using the same gateway group for failover. The DynDNS host will update the IP address as seen by the far side, so that the remote endpoint will know to accept traffic from the newly activated WAN.

Wireless Internal Protection

When configuring IPsec to add encryption to a wireless network, as described in *Additional protection for a wireless network*, choose the OPT interface which corresponds to the wireless card. When using an external wireless access point, pick the interface which is connected to the wireless access point.

Phase 1 Settings

The settings here control the phase 1 negotiation portion of the tunnel, as described previously.

General Information

Disabled Controls whether or not this tunnel (and its associated phase 2 entries) are active and used.

Key Exchange Version This can be *IKEv1*, *IKEv2*, or *Auto*. The differences are discussed in *IKE*.

IKEv1 IKEv1 is more common and widely supported, but has known issues with supporting common modern issues such as dealing with NAT or mobile clients.

IKEv2 (Default) An updated version of the protocol which has increased capabilities and security, as well as built-in support for mobile clients and NAT.

Note: IKEv2 is the best choice when supported by both endpoints.

Auto This option uses IKEv2 when initiating, but will accept either IKEv2 or IKEv1 when responding.

Internet Protocol The protocol for the *outside* of the tunnel. That is, the protocol that will be used between the outside peer addresses. For most, this will be *IPv4* , but if both ends are capable of IPv6, that may be used instead. Whichever protocol is chosen here will be used to validate the Remote Gateway and the associated identifiers.

Note: A tunnel using IKEv1 can only carry the same protocol traffic in Phase 2 as was used for Phase 1. For example, IPv4 peer addresses restrict Phase 2 to IPv4 networks only. A tunnel using IKEv2 can carry both IPv4 and IPv6 traffic at the same time in Phase 2 no matter which protocol was used for Phase 1.

Interface This determines which part of the network will be the termination point (end point) for the IPsec tunnel. If the tunnel will be connecting to a remote server, then WAN is likely the desired setting. This can also be a virtual IP address. A gateway group can also be used for automatic failover. See *Interface Selection* earlier in this document for details on selecting the appropriate interface.

Remote Gateway The IP Address for the peer to which the tunnel will be established. This is most likely the WAN IP address of the remote firewall.

This may be set to an IP address or a fully qualified domain name. When set to use a name, the entry is periodically resolved by DNS and updated when a change is detected.

To allow connections from any endpoint, use `0.0.0.0/0` for IPv4 or `::` for IPv6. When allowing connections from any remote endpoint, the **Child SA Start Action** must be set to *None*, and the **Peer Identifier** cannot be set to *Peer IP Address*

Description It is a good practice to keep notes about the tunnel. Enter a few words to describe the purpose of this VPN tunnel, or about the remote end of the tunnel. This serves as a reminder for anyone managing the firewall (present or future) as to who or what will be using the tunnel.

Authentication Method An IPsec phase 1 can be authenticated using a pre-shared key (PSK) or certificates, the **Authentication Method** selector chooses which of these methods will be used for authenticating the remote peer. Fields appropriate to the chosen method will be displayed on the phase 1 configuration screen.

Mutual PSK The peer is validated using a pre-defined string known to both endpoints. Since it is simple a string, there is a possibility that it can be guessed. For this reason a long/complex key is more secure when using PSK mode.

Mutual Certificate Select a CA and certificate used to verify the peer. During the phase 1 exchange, each peer sends its certificate to the other peer and then validates it against a CA. The CA and certificate must be created for the tunnel before attempting to setup the phase 1.

See also:

IPsec Site-to-Site VPN Example with Certificate Authentication

Mutual Certificate (PKCS#11) Similar to *Mutual Certificate* but the certificate is read from a locally attached PKCS#11 device.

Mutual PSK+Xauth Used with mobile IPsec and IKEv1, this selection enables xauth username and password verification along with a shared (or "group") pre-shared key.

Mutual Certificate+Xauth Used with mobile IPsec and IKEv1, this selection enables xauth username and password verification along with certificate authentication using certificates on both the client and server.

Hybrid Certificate+Xauth Used with mobile IPsec and IKEv1, this selection enables xauth username and password verification along with a certificate only on the server side. It is not quite as secure as *Mutual Certificate+Xauth*, but it is easier on the clients.

EAP-TLS Used with mobile IPsec and IKEv2, EAP-TLS verifies that certificates on the client and server are from the same shared CA, similar to *Mutual Certificate*. The client and server certificates require special handling:

- The server certificate must have the firewall hostname as it exists in DNS listed in its Common Name, and again as a Subject Alternative Name (SAN). The firewall IP address must also be listed in a SAN.
- The identifier in Phase 1 must also be set to match the firewall hostname as listed in the Common Name of the certificate.
- The client certificate must have the username listed as the common name and then again as a SAN.

The CA and server certificates must be generated before attempting to configure EAP-TLS. The CA and user certificate must be imported into the client.

EAP-RADIUS Used with mobile IPsec and IKEv2, this selection performs CA verification along with username and password authentication via RADIUS. A RADIUS server must be selected on the **Mobile Clients** tab. Though user certificates are not necessary, EAP-RADIUS still requires that a CA and server certificate be present using the same attributes mentioned under *EAP-TLS*. The CA must be imported to the client, but no user certificate.

EAP-MSCHAPv2 Used with mobile IPsec and IKEv2, EAP-MSCHAPv2 works identically to EAP-RADIUS except the usernames and passwords are defined on the **Pre-Shared Key** tab under **VPN > IPsec** with the **Secret type** set to EAP. It also requires a CA and server certificate with the same requirements listed previously. The CA must be imported to the client, but no user certificate.

Negotiation Mode (IKEv1 only) The type of authentication security used by this tunnel. This can be either **Main** or **Aggressive**.

Main *Main* is the most secure mode, though it also requires more packets between the peers to accomplish a successful negotiation. It is also much more strict in its validation. This mode is best for security, but not speed.

Aggressive *Aggressive* is generally the most compatible and is the fastest mode. It is more forgiving with identifier types, and tends to be more successful when negotiating with third-party devices. It is faster because it sends all of the identifying information in a single packet, which also makes it less secure because the verification of that data is not as strict as that found in main mode.

Identifiers Identifiers are used by IPsec to identify remote peers and associate specific peers with a tunnel and its related settings, such as authentication components (keys, certificates, etc).

My Identifier Identifier type and value sent by this firewall to the far side. It is best left at *My IP Address* and the firewall will fill it in as needed. In some cases an FQDN or similar may be entered so that the value is constant. So long as both sides agree on the identifier, it will work.

Peer Identifier Identifies the remote peer on the other side of the tunnel. It is best left at *Peer IP Address* and the firewall will fill it in as needed. In some cases an FQDN or similar may be entered so that the value is constant. So long as both sides agree on the identifier, it will work.

Identifier Types

My IP Address / Peer IP address This choice is a macro that will automatically use the IP address on the interface, or the selected VIP, as the identifier. For peers, this is the IP address from which incoming IPsec packets are received by this firewall, which should be the **Remote Gateway**.

IP Address The *IP Address* option allows a specific manual IP address to be used as the identifier. One potential use for this would be if the firewall is behind a router performing NAT. The real external IP address could be used in this field.

Distinguished Name A *Distinguished Name* is another term for a fully qualified domain name (FQDN), such as `host.example.com`. Enter a value in that format into the box.

User Distinguished Name A *User Distinguished Name* is an e-mail address, such as `vpn@example.com`.

ASN.1 Distinguished Name If using *Mutual Certificate* authentication, this can be the subject of the certificate being used, or a similar string.

KeyID Tag An arbitrary string to use as the identifier.

Dynamic DNS A hostname to resolve and use as the identifier. This is mostly useful if the firewall is behind NAT and has no direct knowledge of its external IP address aside from a dynamic DNS hostname. This is only available for **My identifier**, not the **Peer identifier**.

Any In cases when the remote identifier is unknown or cannot be matched, the **Peer Identifier** may be set to **Any**. This is more common on certain types of mobile configurations, but it is a much less secure choice than matching the identifier properly.

Pre-Shared Key (*Mutual PSK* authentication only) This key must be exactly the same on both VPN peers. It is case sensitive. Think of this like a "password" for the tunnel. Since this only gets entered once on each side and there is no need to remember it, the best practice is to make this as long and complex as possible.

Click  **Generate new Pre-Shared Key** to populate the field with a random long string suitable for use as a Pre-Shared Key.

> **Warning:** This Pre-Shared Key must be as random as possible to protect the contents of the tunnel.

My Certificate (*Mutual Certificate* authentication only) Defines the certificate which identifies this firewall. The CA which signed this certificate must be copied to the peer. If one is not shown, create or import it under **System > Cert Manager** on the **Certificates** tab.

Peer Certificate Authority (*Mutual Certificate* authentication only) Defines the CA which has signed the certificate sent by the peer. This is used to validate the peer certificate. If it does not show in the list, import it under **System > Cert Manager** on the **Certificate Authorities** tab.

Phase 1 Encryption algorithms

There are many options for encryption algorithms on both phase 1 and phase 2.

Encryption choices depend on the device to which the tunnel will connect and the hardware. Generally speaking, AES-GCM is the most desirable cipher. When connecting to third party devices, AES with a 128-bit key length is a common choice on endpoints which lack support for AES-GCM.

Multiple combinations of these options can be defined using the  **Add Algorithm** button to add another line. The order of the entries is the order of preference, so configure the strongest settings first.

Encryption Algorithm If both sides support AES-GCM, use *AES128-GCM* with a *128* bit **Key Length**. This will combine strong encryption and hashing together and can be accelerated by AES-NI. Failing that, use *AES* With a **Key Length** of *128*. If the peer does not support any of these, use the strongest available option supported by the peer.

Hash Algorithm Hash algorithms are used with IPsec to verify the authenticity of packet data and as a Pseudo-Random Function (PRF). These hash algorithms may also be referred to with HMAC (Hash Message Authentication Code) in the name in some contexts, but that usage varies depending on the hardware or software in use.

When using AES-GCM, this is used solely as a PRF because AES-GCM already performs hashing internally. The best choice for use with AES-GCM is *AES-XCBC*. If a different type of **Encryption Algorithm** is in use, then use *SHA256* if possible. If the peer does not support any of these, use the strongest available option supported by the peer.

PRF Specifically set a manual PRF different than the one the IPsec daemon would choose automatically based on the **Hash Algorithm**. This control is hidden by the GUI unless **PRF Selection** is enabled in the **Advanced Options** section at the bottom of the page.

DH Key Group The best practice is to use DH Group *14 (2048 bit)* or higher if both sides support it. Avoid using groups 1, 2, 22, 23, and 24 as they do not provide sufficient security. As with the other options, if the suggested value is not supported by the peer, use the strongest available option.

Expiration and Replacement

The total lifetime for Phase 1 defines how often the connection will be rekeyed or reauthenticated by the IPsec daemon, in seconds.

Warning: Take care when crafting these values. Incorrect or sub-optimal values can lead to problems such as tunnels failing to renegotiate in a timely manner or multiple duplicate security associations.

See *Troubleshooting Duplicate IPsec SA Entries* for more details.

The specific values for these fields depend on the IKE mode and which mechanisms are supported by both endpoints, but in most cases setting the **Life Time** value will allow the firewall to choose the best option.

Life Time The hard IKE SA life time, in seconds, after which the IKE SA will be expired. This value **must** be larger than **Rekey Time** and **Reauth Time** and cannot be set to the same value.

`28800` total seconds is a good balance of frequent rekeying without being too aggressive.

Tip: Set one endpoint to this recommended value, but use a higher **Life Time** on the other endpoint by at least 10% (e.g. `31680`) to help avoid overlap.

If left empty, the value defaults to 110% of **Rekey Time** or **Reauth Time**, whichever is higher.

Rekey Time Time, in seconds, before the IPsec daemon attempts attempts to establish a new set of keys for the IKE SA. Only supported by IKEv2, and is the best choice for use with IKEv2.

Rekey works without interruption and allows both endpoints to seamlessly change to new keys on the fly. This is optimal, but implementation quality varies by vendor.

Leave blank to automatically calculate the value based on 90% of **Life Time**, or enter a value of `0` to disable rekeying.

Normally both sides will rekey as needed, but if the tunnel often fails when a rekey event occurs, try disabling this feature on one side.

Note: Some clients, especially Windows clients behind NAT, misbehave when they receive a rekey request. In those cases it is safer to allow the client to initiate the rekey by disabling the option on the server.

Reauth Time Time, in seconds, before an IKE SA is torn down and recreated from scratch by the IPsec daemon, including authentication. Supported by IKEv1 and IKEv2, but should be avoided with IKEv2 where possible.

This process can be disruptive to traffic flow unless all peers support IKEv2 make-before-break (*Advanced IPsec Settings*) and overlapping IKE SA entries.

Leave blank to automatically calculate the value based on 90% of **Life Time**, or enter a value of `0` to disable reauthentication.

Rand Time Introduces randomness into the rekey or reauthentication process to avoid both endpoints attempting to renegotiate simultaneously.

A random value up to this amount will be subtracted from **Rekey Time** or **Reauth Time** for each scheduled renegotiation to reduce the chance of collisions.

If left empty, the value defaults to 10% of **Life Time**. Enter `0` to disable randomness.

Warning: Disabling **Rand Time** increases the likelihood of simultaneous renegotiation, which can lead to duplicate security associations.

See *Troubleshooting Duplicate IPsec SA Entries* for more details.

Advanced Options

Child SA Start Action This option forces specific behavior performed by the IPsec daemon loading a phase 2 configuration ("Child SA") during service startup. This happens at boot and when the service is restarted for any reason.

Default Automatically chooses behavior based on other settings.

None (Responder Only) When set, then IPsec daemon will not attempt to initiate the tunnel. The tunnel will only be established by an initiation attempt from the far side. Also, if DPD detects that the tunnel has failed, the tunnel will be left down rather than restarted, leaving it up to the far side to reconnect.

This is the default behavior for mobile IPsec and tunnels with unknown remote endpoints.

Initiate at Start (VTI or Tunnel Mode) When set, the firewall will attempt to establish the IPsec tunnel immediately when the IPsec daemon starts.

This is the default behavior for VTI mode.

Initiate on Demand (Tunnel mode only) When set, traffic which matches the networks in phase 2 definitions for this tunnel ("interesting traffic") will trigger initiation of the tunnel.

This is the default behavior of Tunnel Mode.

Note: Finding "interesting" traffic relies on trap policies to function, and trap policies are not compatible with VTI mode.

Child SA Close Action Controls how the IPsec daemon behaves when a child SA (P2) is unexpectedly closed by the peer.

Default Retains the default behavior based on other settings for the tunnel.

Close connection and clear SA Removes the child SA and does not attempt to establish a new SA. This is the desired behavior when acting in a **Responder Only** or mobile IPsec role.

Restart/Reconnect Immediately attempts to reconnect the child SA. This ensures that the tunnel reestablishes properly in cases that do not support trap policies, such as routed IPsec (VTI). Set this on one side only if the tunnel does not reconnect after it disconnects, rekeys, or reauthenticates.

> **Warning:** This option **must not** be set on both peers! Otherwise both peers will attempt to initiate and hold open multiple copies of each child SA.

Close connection and reconnect on demand Clears the child SA and reinstalls trap policies to watch for interesting traffic. Will reestablish the tunnel on demand when traffic attempts to cross the tunnel.

This option is not compatible with modes which do not support trap policies, such as routed IPsec (VTI).

NAT Traversal (IKEv1 Only) Also known as NAT-T. NAT Traversal encapsulates ESP traffic for IPsec inside of UDP packets, to more easily function in the presence of NAT. If this firewall or the firewall on the other end of the tunnel is behind a NAT device, then NAT Traversal will likely be necessary for the tunnel to function properly.

Auto (Default) Allows the IPsec daemon to detect and use NAT Traversal automatically when it determines one or both peers is behind NAT.

Force Instructs the IPsec daemon to always use NAT Traversal for the tunnel. This can help if there is a known issue detecting NAT, or with issues carrying ESP traffic between the two endpoints even when neither side is behind NAT.

IKEv2 integrates NAT Traversal natively so the option is unnecessary in that case.

MOBIKE An extension to IKEv2 which handles multi-homed clients and clients which roam between different IP addresses. This is primarily used with mobile clients to allow them to switch remote addresses while keeping the connection active. Leave disabled unless the remote peer must change addresses dynamically.

Gateway Duplicates When set, the GUI validation allows multiple Phase 1 configurations to the same remote endpoint.

> **Warning:** This option also disables automatic static routes to the peer via specific WAN gateways. Traffic will follow the default route, not the tunnel interface, unless manual static routes redirect the traffic.

Split Connections (IKEv2 Only) When an IKEv2 tunnel has multiple Phase 2 definitions, by default the settings are collapsed in the IPsec configuration such that all P2 combinations are held in a single child SA.

Split Connections changes this behavior to be more like IKEv1 where each P2 is its configured by the daemon as own separate child SA.

Certain scenarios require this behavior, such as:

- The remote peer does not properly handle multiple addresses in single traffic selectors. This is especially common in Cisco, Checkpoint, Fortinet, and Juniper equipment.
- Each child SA must have unique traffic selector or proposal settings. This could be due to the peer only allowing specific combinations of local/remote subnet pairs or different encryption options for each child SA.

PRF Selection When set, the GUI enables a control to specifically set a Pseudo-Random Function (PRF) rather than allow the IPsec daemon to choose one automatically based on the selected Hash Algorithm. Can be useful in combination with AEAD encryption algorithms such as AES-GCM.

Custom IKE/NAT-T Ports In rare situations the remote endpoint may be running IPsec on alternate port numbers for IKE and NAT-T. These settings can accommodate such endpoints.

Remote IKE Port The UDP port for IKE on the remote gateway. Leave empty for the default automatic behavior (Port `500` for IKE and `4500` for NAT-T)

Remote NAT-T Port The UDP port for NAT-T on the remote gateway.

Note: If **Remote IKE Port** is empty and NAT-T contains a value, the tunnel will use only NAT-T.

Dead Peer Detection Dead Peer Detection (DPD) is a periodic check that the host on the other end of the IPsec tunnel is still alive. This detects when an IPsec peer has lost connectivity or otherwise is unreachable. If a DPD check fails, the tunnel is torn down by removing its associated SAD entries and a fresh negotiation is attempted.

The default settings are sufficient for most connections. Increase the values for bad quality links to avoid tearing down a usable, but lossy, tunnel.

Delay Time between DPD probe attempts. The default of `10` is best.

Max Failures Number of failures before the peer is considered down. The default of `5` is best.

Note: The default values of `10` seconds and `5` failures will result in the tunnel being considered down after approximately one minute.

Phase 2 Settings

The phase 2 settings for an IPsec tunnel govern how the tunnel handles traffic (e.g. policy-based or route-based, see *IPsec Modes*) as well as the encryption of that traffic.

Phase 2 entries are used in a few different ways, depending on the IPsec configurations:

- For policy-based IPsec tunnels, this controls which subnets will enter IPsec. Multiple phase 2 definitions can be added for each phase 1 to allow using multiple subnets inside of a single tunnel.
- For route-based IPsec, this controls the VTI interface addresses.
- For mobile IPsec this primarily controls the encryption for phase 2, but can also optionally be used by the IPsec daemon or export utilities to generate a list of networks to the clients for use in split tunneling.

Each Phase 2 entry has the following options:

Disabled An on/off switch for this Phase 2 entry only.

Mode The IPsec Mode for this Phase 2 entry, which controls how the tunnel handles traffic. See *IPsec Modes* for more detailed explanations of each type of mode.

With policy-based IKEv1 tunnels, this must match the outer protocol of the tunnel, for example an IPv4 peer would be Tunnel IPv4. Policy-based IKEv2 tunnels can have either/or (or both).

Tunnel IPv4 A policy-based tunnel that will carry traffic between IPv4 networks matching the specified **Local Network** and **Remote Network**.

Tunnel IPv6 A policy-based tunnel that will carry traffic between IPv6 networks matching the specified **Local Network** and **Remote Network**.

Transport Encrypts all traffic between the endpoints. The **Local Network** and **Remote Network** are not set for transport mode, it assumes the addresses based on the phase 1 settings.

Routed (VTI) Routed IPsec using Virtual Tunnel Interfaces. The **Local Network** and **Remote Network** define the addresses used by the firewall for the VTI interface. Typically only one Phase 2 entry is present for each address family (e.g. one for IPv4, one for IPv6)

See *Routed IPsec (VTI)* for more information.

Local Network

Tunnel Mode Defines which subnet or host can be accessed from the other side of the VPN tunnel. This is typically the LAN or other internal subnet for the VPN, but can also be a single IP address if only one client needs to use the tunnel. The **Type** selector is pre-loaded with choices for each interface (e.g. *LAN subnet*), as well as *Address* and *Network* choices that allow entering an IP address or subnet manually.

Most often this is set to *LAN subnet*, meaning the entire LAN will be accessible from the remote network.

NAT/BINAT Sets a *different* subnet or address which is used by IPsec to perform NAT on the local network addresses to make them appear to the remote peer as a different subnet.

Set to *None* to disable NAT for the tunnel.

For more details, see *NAT with IPsec Phase 2 Networks*.

Routed (VTI) Mode Sets the local IP address and subnet mask of the `ipsecX` interface.

Remote Network

Tunnel Mode (Non-mobile only) Specifies the IP Address or Network which exists on the other (remote) side of the VPN. This field operates similarly to the **Local Network** option.

Routed (VTI) Mode Sets the remote IP address for the `ipsecX` interface tunnel network (the remote address of the VTI).

Description A description for this Phase 2 entry. Shows up in the IPsec status for reference.

Phase 2 Proposal (Child SA)

Protocol Controls how IPsec protects its traffic.

ESP (Encapsulating Security Payload) Encrypts traffic before sending it to the peer.

In nearly all circumstances, ESP is the correct choice.

AH (Authenticated Header) Provides assurance the traffic came from a trusted source but does not provide encryption. Rarely used in practice.

Note: With automatic VPN rules (*Disable Auto-added VPN rules*), the firewall automatically passes the appropriate ESP or AH protocol traffic from the remote endpoint. If automatic VPN rules are disabled, add manual rules to pass the traffic instead.

Encryption algorithms Sets the encryption algorithms used when negotiating Phase 2 child SA entries with peers. Must match values available to and configured on the peer.

In systems with AES-NI, the fastest and most secure choice is AES-GCM if it is supported by the peer. When using AES-CGM do not select any options for **Hash Algorithms** in Phase 2.

If AES-NI cannot be used both both peers, use AES with a 128-bit or higher key length.

This set of controls allows for multiple selections so that multiple choices will be accepted when acting as a responder, or proposed when working as an initiator. The best practice is to only select a single desired cipher on both peers, but in some cases, such as mobile clients, selecting multiple will allow a tunnel to work better in both a responder and initiator role.

Hash algorithms Controls which hash algorithms are used when negotiating phase 2 child SA entries with peers. Must match values available to and configured on the peer.

As with the Encryption Algorithms, multiple hashes may be selected. The best practice is to select a single desired choice if possible. For more discussions on the quality of the various hash types, see *Phase 1 Settings*.

The optimal choice for speed and security is **SHA256**, unless using AES-GCM for the **Encryption Algorithm**. With AES-GCM, no **Hash Algorithm** should be selected as AES-GCM performs hashing on its own.

PFS key group Perfect Forward Secrecy (PFS) provides keying material with greater entropy, hence improving the cryptographic security of the connection, at the cost of higher CPU usage during rekeying.

The options have the same properties as the DH key group option in phase 1 (See *DH key group*), and some products also refer to them as "DH" values even in Phase 2.

The optimal choice for speed and security is *14 (2048 bit)*. The default is *off*.

Life Time The hard child SA life time, in seconds, after which the child SA will be expired. This value **must** be larger than **Rekey Time** and cannot be set to the same value.

`3600` total seconds is a good balance of frequent rekeying without being too aggressive.

Tip: Set one endpoint to this recommended value, but use a higher **Life Time** on the other endpoint by at least 10% (e.g. `5400`) to help avoid overlap.

If left empty, the value defaults to 110% of **Rekey Time**. If both **Life Time** and **Rekey Time** are empty, defaults to `3960`.

Rekey Time Time, in seconds, before the child SA establishes a new set of keys. This works without interruption and allows both endpoints to seamlessly change to new keys on the fly.

Leave blank to automatically calculate the value based on 90% of **Life Time**. If both **Life Time** and **Rekey Time** are empty, defaults to `3600`. Enter a value of `0` to disable rekeying.

Note: If rekeying is disabled, connections can be interrupted while a new child SA is negotiated after an old entry expires.

Rand Time Introduces randomness into the rekey process to avoid both endpoints attempting to renegotiate simultaneously.

A random value up to this amount will be subtracted from **Rekey Time** for each scheduled renegotiation to reduce the chance of collisions.

If left empty, the value defaults to 10% of **Life Time**. Enter `0` to disable randomness.

Warning: Disabling **Rand Time** increases the likelihood of simultaneous renegotiation, which can lead to duplicate security associations.

See *Troubleshooting Duplicate IPsec SA Entries* for more details.

Automatically ping host For use on non-mobile tunnels, this option tells the firewall to initiate a ping periodically to the specified IP address. This option only works if the firewall has an IP address inside of the **Local Network** for this Phase 2 entry and the value of the ping host here must be inside of the **Remote Network**.

See *Configuring IPsec Keep Alive* for additional information.

19.8.2 IPsec Tunnel List

The IPsec page located at **VPN > IPsec** allows management of IPsec VPN tunnels. A brief summary of existing tunnel settings is also displayed on this page.

Each IPsec tunnel will have one phase 1 definition, and one or more phase 2 definitions.

Phase 1 definitions handle how the tunnel connects to the remote peer. This includes the remote gateway, authentication information such as identifiers and pre-shared keys or certificates, NAT Traversal and DPD settings.

After adding a phase 1 definition, click the larger  button underneath a phase 1 entry to display and manage its phase 2 entries.

Phase 2 definitions handle how local/internal networks are sent across a tunnel. Multiple local subnets (or individual hosts) can be used on a single IPsec tunnel by adding multiple Phase 2 entries.

Phase 2 definition settings include the local and remote networks for traffic which will traverse the tunnel, and phase 2 encryption proposal settings.

See also:

IPsec - All other IPsec articles.

19.8.3 NAT with IPsec Phase 2 Networks

pfSense® software supports for NAT on policy-based IPsec Phase 2 entries to make the local network appear to the remote peer as a different subnet or address. This can be used to work around subnet conflicts or connect to vendors without renumbering a local network.

Warning: NAT is not currently compatible with route-based VTI IPsec tunnels.

Configuration

NAT is configured by the **NAT/BINAT Translation** options on an IPsec Phase 2 entry in tunnel mode, in combination with the **Local Network** settings.

Local Network Values of **Type** and **Address** specify the **actual** local network (e.g. LAN subnet).

NAT/BINAT Translation Values of **Type** and **Address** specify the **translated** network visible to the far side.

NAT Types

There are two main modes for NAT with IPsec:

Binat - 1:1 NAT When both the actual and translated local networks use the same subnet mask, the firewall will directly translate the networks to one another inbound and outbound. Can also be used for single addresses.

This allows remote host to directly contact local hosts using their equivalent NAT addresses, provided that IPsec rules allow the traffic to pass.

NAT - Overload/PAT Style If the Local Network is a subnet, but the **NAT/BINAT Translation** address is set to a single IP address, then a 1:many NAT (PAT) translation is set up that works like an outbound NAT rule on WAN. All outbound traffic will be translated from the local network to the single IP address in the NAT field.

Note: Inbound traffic from the remote network to individual local hosts is not possible in this mode.

Warning: NAT+IPsec cannot be configured between two different sized subnets (e.g. It cannot NAT a /24 subnet to a /27 subnet).

Example

Consider an IPsec tunnel to a Vendor which requires `172.16.5.0/24` for the network on this firewall. However, the LAN is actually `192.168.1.0/24`, and renumbering is not feasible.

To accommodate this scenario, set the Phase 2 values as follows:

Local Network

Type *Network*

Address `192.168.1.0/24`

NAT/BINAT Translation

Type *Network*

Address `172.16.5.0/24`

Firewall Rules

NAT is processed before firewall rules, so firewall rules on the IPsec tab refer to the network in **Local Network**.

Remote End Notes

The far side of the tunnel does not need any knowledge of the actual **Local Network**. Their tunnel is built between their local network and the **NAT/BINAT Translation** value.

Packet Capturing Quirk

In a packet capture, the **Local Network** addresses are shown on outbound traffic, not the translated address. This does not indicate any problem.

19.8.4 Routed IPsec (VTI)

Route-based IPsec is an alternative method of managing IPsec traffic. It uses `if_ipsec(4)` from FreeBSD 11.1+ for Virtual Tunnel Interfaces (VTI) and traffic is directed using the operating system routing table. It does not rely on strict kernel security association matching like policy-based (Tunneled) IPsec.

A routed IPsec tunnel creates an `ipsecXXXX` interface at the operating system level and this interface has its own IP address. The `ipsecXXXX` interface must be assigned so it can be used for purposes such as static or dynamic routing, daemon binding, traffic monitoring, and so on.

Once assigned, the IPsec interface also gains an automatic gateway which provides policy routing and gateway group capabilities.

Note: Routed IPsec is not replacing traditional tunneled IPsec, both may be used. The choice is up to the user when creating an IPsec Phase 2 entry.

Note: Routed IPsec is available on firewalls running pfSense® software version 2.4.4-RELEASE or later.

See also:

The Hangouts Archive contains a video which covers Routed IPsec.

Prerequisites

First, pick a transit network. This is similar to choosing a tunnel network for an OpenVPN instance. Typically this is a /30 network in an unused subnet. This example uses `10.6.106.0/30`.

IPsec Configuration

- Create an IPsec Phase 1 entry as usual.
- Create a Phase 2 entry under this Phase 1, set with...
  - Set **Mode** to *Routed (VTI)*
  - Enter `10.6.106.1` for the **Local Network Address**
  - Enter `10.6.106.2` for the **Remote Network Address**
  - Add a useful **Description**
  - Set the **Proposal** settings as needed
- Click **Save**, then click **Apply Changes**

IPsec Interface Assignment

- Navigate to **Interfaces > Assignments**
- Pick the new `ipsecX` interface from the **Available Network Ports** list
- Click **+ Add**
- Note the new interface name, e.g. *OPT1*
- Navigate to **Interfaces > [New Interface Name]**
- Check **Enable**
- Give the interface a more suitable name using the **Description** field (e.g. VTI_FOO)
- Leave the **IPv4 Configuration Type** and **IPv6 Configuration Type** set to *None*
- Click **Save**, then click **Apply Changes**

A gateway is created automatically and can be used for static routing, policy routing, and so on.

At this point the interface is available for use like any other interface. It can be used for packet captures, traffic graphs, binding daemons, routing protocols, and other tasks never before possible with IPsec on pfSense software!

Routing

Until routing is configured, no traffic will attempt to cross the IPsec tunnel except for gateway monitoring probes, if they are enabled.

Static Routes

To setup static routes, navigate to **System > Routing**, **Static Routes** tab. Add a new route there using the assigned IPsec interface gateway.

Policy Routes

To policy route traffic across a routed IPsec tunnel, use the assigned IPsec interface gateway in firewall rules as usual for policy routing.

See also:

Policy Routing Configuration

Dynamic Routes

The assigned IPsec interface can be used by the FRR dynamic routing daemon. BGP and OSPF can both operate across routed IPsec interfaces.

Routed IPsec Firewall Rules

Routed IPsec traffic appears to the OS on both the specific IPsec interface and the enc0 interface, which is governed by the rules on the **IPsec** tab. Though a tab appears for the assigned interface, **traffic must be passed on the IPsec tab**.

Caveats

Routed IPsec works best when both sides support routed IPsec. It can still work when only one side supports routed IPsec, but most of its benefits are lost.

Rather than managing IPsec Phase 2 entries, routes must be managed instead. Since this can be automated with dynamic routing protocols this is not a large concern.

Firewall rule processing can be confusing, as mentioned in *Routed IPsec Firewall Rules*. This is still undergoing testing, but likely means that reply-to will not function. There are also known issues with NAT, notably that NAT to the interface address works but 1:1 NAT or NAT to an alternate address does not work.

19.8.5 IPsec and firewall rules

When an IPsec tunnel is configured, pfSense® automatically adds hidden firewall rules to allow UDP ports 500 and 4500, and the ESP protocol from the **Remote gateway** IP address destined to the Interface IP address specified in the tunnel configuration. When mobile client support is enabled the same firewall rules are added except with the source set to *any*. To override the automatic addition of these rules, check **Disable all auto-added VPN** rules under **System > Advanced** on the **Firewall/NAT** tab. When that box is checked, firewall rules must be manually added for UDP 500, UDP 4500, and ESP to the appropriate WAN interface.

Traffic initiated from the remote end of an IPsec connection is filtered with the rules configured under **Firewall > Rules** on the **IPsec** tab. Here restrictions may be placed on resources made accessible to remote IPsec users. To control what traffic can be passed from local networks to the remote IPsec VPN connected devices or networks, the rules on the local interface where the host resides control the traffic (e.g. connectivity from hosts on LAN are controlled with LAN rules).

19.8.6 Using IPsec with Multiple Subnets

On current versions of pfSense® software, additional subnets are handled by adding an additional Phase 2 entry to cover the path to pass through the tunnel.

For example, for 172.16.0.0/24 and 172.16.1.0/24 at Site A, and 10.0.0.0/24 at Site B, define two Phase 2 entries on both sides:

On the Site A Firewall:

```
172.16.0.0/24 to 10.0.0.0/24
172.16.1.0/24 to 10.0.0.0/24
```

On the Site B Firewall:

```
10.0.0.0/24 to 172.16.0.0/24
10.0.0.0/24 to 172.16.1.0/24
```

This works for any additional networks on either side (VPN subnets, networks on the other end of VPNs connected to the remote router, etc).

If the equipment to which the tunnel connects does not support multiple Phase 2's, it may be necessary to employ supernetting/CIDR summarization (See below) to fit the networks into a single Phase 2.

Supernetting Example

At Site A, there is one subnet, 10.0.0.0/24. This should reach 192.168.0.0/24, 192.168.1.0/24, and 192.168.2.0/24 at Site B.

Due to the "closeness" of the subnets, they could be grouped into a larger network in the tunnel definition: 192.168.0.0/22 (This would also include 192.168.3.0/24)

19.8.7 Advanced IPsec Settings

The **Advanced Settings** tab under **VPN > IPsec** contains options to control, in general, how the IPsec daemon behaves and how traffic is handled with IPsec.

IPsec Logging Controls These options control which areas of the IPsec daemon generate log messages and their level of detail. For information on viewing the log, see *IPsec Logs*.

In most cases the optimal settings are the default: **IKE SA**, **IKE Child SA**, and **Configuration Backend** set to *Diag*, and all others set to *Control*.

Configure Unique IDs as Controls how the IPsec daemon treats new connections with an identifier which matches an existing connection. In most cases a new connection is intended to replace an older connection, but certain use cases such as mobile clients may require multiple connections from the same remote identifier.

Yes (Replace) The new connection is accepted by the IPsec daemon and it replaces the old connection, which is disconnected.

No The new connection is accepted and the old connection is replaced only if the peer sends an `INITIAL_CONTACT` notification.

Never The new connection is always allowed, and `INITIAL_CONTACT` notifications are ignored.

Keep The new connection is rejected and the old connection remains active.

IPsec Filter Mode **Experimental.** Controls how the firewall will filter IPsec traffic.

Filter IPsec Tunnel and VTI on IPsec tab (enc0) The default behavior. Rules on the IPsec tab filter all IPsec traffic, including both tunnel mode and VTI mode.

This is limited in that it **does not** allow for filtering on assigned VTI interfaces, and does not allow for NAT or `reply-to` to function for VTI rules.

Filter IPsec VTI on assigned interfaces, block all tunnel mode traffic Enables firewall rules for assigned VTI interfaces, NAT on VTI interfaces, and `reply-to` for rules on assigned VTI interface tabs. However, when set to filter on assigned VTI interfaces, **all tunnel mode traffic is blocked**.

> **Warning:** Do not set this option unless all IPsec tunnels are using VTI. This is incompatible with mobile IPsec as it is only capable of using tunnel mode.

IP Compression Propose support for IPComp compression.

> **Warning:** Though the option is present in the GUI, the underlying operating system does not yet fully support IP compression.

Strict Interface Binding When set, the IPsec daemon configuration binds only to the interfaces required by the configuration, rather than binding to all interfaces.

This option is more secure but is known to break with interfaces which have dynamic IP addresses. Only enable this option in environments where it has been lab tested and proven to work as intended.

Unencrypted Payloads in IKEv1 Main Mode Some IPsec implementations send the third Main Mode message unencrypted, probably to find the PSKs for the specified ID for authentication. This is similar to Aggressive Mode, and has the same security implications: A passive attacker can sniff the negotiated Identity, and start brute forcing the PSK using the HASH payload. The best practice is to keep this option disabled unless the implications are fully understood and compatibility to such devices is required (for example, some SonicWall devices).

Maximum IKEv1 Phase 2 Exchanges IKEv1 phase 2 rekeying for one VPN gateway can be initiated in parallel. By default only `3` parallel rekeys are allowed. Undersized values can break VPN connections with many phase 2 definitions. If unsure, set this value to match the largest number of phase 2 entries on any phase 1.

MSS Clamping Enable maximum segment size clamping on TCP flows over IPsec tunnels. This helps overcome problems with path MTU discovery (PMTUD) on IPsec VPN links.

This is useful is large TCP packets have problems traversing the VPN, or if slow/choppy connections across the VPN are observed by users. Ideally it should be set to the same value on both sides of the VPN, but traffic will have MSS clamping applied in both directions.

Enable When set, the **Maximum MSS** option is available and its value is used by the firewall configuration.

Maximum MSS The maximum segment size set in TCP packets flowing across IPsec VPN tunnels. Defaults to `1400`. Must be low enough to account for the overhead of IPsec and the MTU of the link, but no so low that unnecessarily small segments are sent as that can be inefficient.

Enable Cisco Extensions Enables the Unity plugin which provides support for Cisco Extensions such as `Split-Include`, `Split-Exclude`, and `Split-DNS` for IKEv1 XAuth mobile clients. This allows clients which support these extensions to obtain values automatically when connecting to a mobile IPsec VPN.

Strict CRL Checking When set, the IPsec daemon requires availability of a fresh CRL for peer authentication based on certificate signatures to succeed. Primarily useful when the CRL is obtained dynamically (e.g. OCSP).

> **Warning:** If there is no CRL available for a CA, validation will fail.

Make Before Break Controls whether IKEv2 Reauthentication uses Make-before-Break or Break-before-Make when an IKE Security Association (SA) expires. Must be supported by both peers.

Only relevant for IKEv2 tunnels using reauthentication, it does not affect IKEv1 tunnels or IKEv2 tunnels set to rekey.

Break-before-Make (Unchecked, Default) Deletes IKE and Child SAs before reauthenticating and making a new set of SAs. This behavior is standard and well-supported, but disruptive as there is a small gap between the old and new SA set in which IPsec connectivity is unavailable.

Make-before-Break (Checked) Reauthenticates and makes a new SA set before deleting the old SA set. This eliminates the connectivity disruption, but requires that both endpoints support overlapping IKE and Child SA entries.

Asynchronous Cryptography Allows cryptographic framework jobs to be dispatched in a multi-threaded manner to increase performance. Jobs are handled in the order they are received so that packets will be reinjected in the correct order.

> **Warning:** This option can increase performance, but may be unstable on certain hardware. When enabling this option, test connectivity during a maintenance window to ensure proper behavior. See Bug #8964 for details.

Custom Ports Rare situations may require the firewall to listen for inbound IPsec packets on alternate port numbers for IKE and NAT-T. These settings can accommodate such cases, but affect every tunnel on the firewall.

Leave empty for the default behavior, which is to use UDP port `500` for IKE and `4500` for NAT-T.

Auto-exclude LAN Address Set up an automatic IPsec bypass for traffic to and from the LAN subnet, so it does not get captured by policy-based IPsec.

Additional IPsec Bypass Configures additional manual IPsec bypass behavior. When set, the GUI exposes the **IPsec Bypass Rules** control.

IPsec Bypass Rules Custom rules which allow traffic matching combinations of **Source Address** and **Destination Address** pairs to be excluded from IPsec policies.

Source Address The source address or network to exclude, and its mask.

Destination Address The corresponding destination address or network to exclude, and its mask.

Note: These values are considered *together*. A packet must match **both** the source and destination to bypass IPsec policies.

These rules are useful to exclude traffic between multiple local networks, especially when a policy-based IPsec tunnel is set to use `0.0.0.0/0` as the remote network.

19.8.8 Configuring IPsec Keep Alive

Any IP address within the **Remote Network** of this phase 2 definition may be used. It does not have to reply or even exist, simply triggering traffic destined to that network periodically will keep the IPsec connection up and running.

For this feature to work, the firewall must have an IP address assigned inside the **Local Network**. Otherwise it cannot generate the necessary traffic to bring up the tunnel.

19.8.9 Mobile IPsec

Choosing a Mobile IPsec Style

Currently only one type of mobile IPsec may be configured at a time, though there are multiple different styles to choose from.

- *IKEv2 with EAP-TLS* for per-user certificate authentication
- *IKEv2 with EAP-MSCHAPv2* for local username and password authentication
- *IKEv2 with EAP-RADIUS* for remote username and password authentication
- *Xauth+PSK* for local or remote username and password authentication
- Xauth+RSA for certificates and local or remote username and password authentication
- *Pre-Shared Key* for basic IPsec connectivity from older clients
- *L2TP/IPsec* for local or remote username and password authentication with clients that do not support one of the above methods.

See also:

- *Configuring IPsec IKEv2 Remote Access VPN Clients on Windows*
- *Configuring IPsec IKEv2 Remote Access VPN Clients on Ubuntu*
- *Configuring IPsec IKEv2 Remote Access VPN Clients on Android*
- *Configuring IPsec IKEv2 Remote Access VPN Clients on OS X*
- *Configuring IPsec IKEv2 Remote Access VPN Clients on iOS*

As of this writing, most **current** operating systems natively support IKEv2 or can use an app/add-on. It is currently the best choice, and will be the one demonstrated later in this chapter. Windows 7 and later, MAC OS X 10.11 (El Capitan) and later, iOS 9 and later, and most Linux distributions have support built in for IKEv2. There is a simple-to-use strongSwan IKEv2 app for Android 4.x and later.

Note: All IKEv2 types require a certificate structure including at least a Certificate Authority and a Server Certificate, and in some cases user certificates. For more information on Certificates, see *Certificate Management*. Clients can be very picky about certificate attributes, so pay close attention to this chapter when creating the certificate structure.

Warning: When generating a **Server Certificate** for use with IKEv2, the Common Name of the certificate must be the firewall's name as it exists in DNS. The name must be repeated again as an FQDN type Subject Alternative Name (SAN). The IP address of the firewall must also be present as an IP Address type SAN. This information will be repeated later in the chapter, but requires extra emphasis due to its importance. See *Create a Server Certificate*

IKEv2 with EAP-MSCHAPv2

With support for IKEv2 now widespread, it is an ideal choice for current operating systems. Though there are several variations, EAP-MSCHAPv2 is the easiest to configure since it does not require generating or installing per-user certificates and does not require a working RADIUS server. The CA Certificate must still be installed onto the client as a trusted root certificate.

EAP-MSCHAPv2 allows for username and password authentication using passwords stored on the **Pre-Shared Keys** tab under **VPN > IPsec**. These passwords are stored in plain text, so it is not as secure as using a RADIUS server, though it is more convenient.

See also:

- *IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2*

IKEv2 with EAP-RADIUS

EAP-RADIUS works identically to EAP-MSCHAPv2 except that user authentication happens via RADIUS. When EAP-RADIUS is chosen, a RADIUS server must on the **Mobile Clients** tab. The RADIUS server must accept and understand EAP requests and it must also allow MSCHAPv2. Password security is left up to the RADIUS server.

EAP-RADIUS is typically the best choice when a RADIUS server is available.

See also:

- *IPsec Remote Access VPN Example Using IKEv2 with EAP-RADIUS*

IKEv2 with EAP-TLS

EAP-TLS uses per-user certificate authentication instead of username and password authentication. As such, EAP-TLS requires generating certificates for each user, which makes it a bit more cumbersome from an administration standpoint. Certificates are validated against the CA similar to OpenVPN. The CA certificate, user certificate and its associated key must all be imported to the client properly.

Warning: When creating user certificates, the username must be used as the certificate common name and **again** as a DNS/FQDN type Subject Alternative Name. If the same name is not present in both places, clients may not be validated properly.

See also:

- *IPsec Remote Access VPN Example Using IKEv2 with EAP-TLS*

IKEv1 with Xauth and Pre-Shared Keys

Xauth+PSK works on a majority of platforms, the notable exception being current versions of Android. Windows XP through Windows 8 can use the Shrew Soft client, but Windows 10 does not currently work with any client. OS X and iOS can use their built-in client to connect.

Note: When using Xauth, local users must exist in the **User Manager** and those users must have the *User - VPN - IPsec Xauth Dialin* privilege.

See also:

- *IPsec Remote Access VPN Example Using IKEv1 with Xauth*

IKEv1 with Xauth and RSA Certificates

Xauth+RSA works in most of the same conditions as Xauth+PSK, though it does in fact work on current Android devices. Certificates must be made for each user, and the certificates must be imported into the clients.

IKEv1 with Pre-Shared Keys Only

Pre-Shared Key only IPsec VPNs for mobile IPsec have become rare in modern times. Support was not very common, only found in the Shrew Soft client, some very specific Android versions (such as those from Motorola), and in other third-party clients. They are not very secure, and are no longer recommended for general use. The only time they may be needed is in cases when the far side cannot support any other method.

See also:

- *IPsec Remote Access VPN Example Using IKEv1 with Pre-Shared Keys*

L2TP/IPsec (IKEv1)

L2TP/IPsec is a unique combination that, unfortunately, does not work very well in most cases. In this style of setup, Mobile IPsec is setup to accept Transport Mode connections which secure all traffic between the public IP address endpoints. Across this transport channel, an L2TP connection is made to tunnel user traffic in a more flexible way. Though support for this model is found in most versions of Windows, MAC, Android, and other Operating Systems, they are all picky in different incompatible ways about what will work.

For example, the Windows client does not work properly when the client system is behind NAT, which is the most common place that a VPN client would find itself. The problem is in an interaction between the client and the IPsec daemon used on pfSense®, strongSwan. The strongSwan project states that it is a bug in the Windows client, but it is unlikely to be fixed since both strongSwan and Windows have focused their mobile client efforts on more modern and secure implementations such as IKEv2 instead.

Warning: L2TP/IPsec should be avoided when possible.

See also:

- *L2TP/IPsec Remote Access VPN Configuration Example*

See also:

- *Remote Access Mobile VPN Client Compatibility*

Mobile IPsec allows creation of a so-called "Road Warrior" style VPN, named after the variable nature of anyone who is not in the office that needs to connect back to the main network. It can be a sales person using Wi-Fi on a business trip, the boss from his limo via 3G modem, or a programmer working from their broadband line at home. Most of these will be forced to deal with dynamic IP addresses, and often will not even know the IP address they have. Without a router or firewall supporting IPsec, a traditional IPsec tunnel will not work. In telecommuting scenarios, it's usually undesirable and unnecessary to connect the user's entire home network to the office network, and doing so can introduce routing complications. This is where IPsec Mobile Clients are most useful.

There is only one definition for Mobile IPsec on pfSense®, so Instead of relying on a fixed address for the remote end of the tunnel, Mobile IPsec uses some form of authentication to allow a username to be distinguished. This could be a username and password with IKEv2 and EAP or xauth, or a per-user Identifier and Pre-Shared Key pair, or a certificate.

19.8.10 Testing IPsec Connectivity

The easiest test for an IPsec tunnel is a ping from one client station behind the firewall to another on the opposite side. If that works, the tunnel is up and working properly.

As mentioned in *Accessing Firewall Services over IPsec VPNs*, traffic initiated from the pfSense® firewall will not normally traverse the tunnel without extra routing, but there is a quick way to test the connection from the firewall itself by specifying a source when issuing a ping.

There are two methods for performing this test: the GUI, and the shell.

Specifying a Ping Source in the GUI

In the GUI, a ping may be sent with a specific source as follows:

- Navigate to **Diagnostics > Ping**
- Enter an IP address on the remote router within the remote subnet listed for the tunnel in the **Host** field (e.g. `10.5.0.1`)
- Select the appropriate **IP Protocol**, likely *IPv4*
- Select a **Source Address** which is an interface or IP address on the local firewall which is inside the local Phase 2 network (e.g. Select *LAN* for the LAN IP address)
- Set an appropriate **Count**, such as the default *3*
- Click **Ping**

If the tunnel is working properly, ping replies will be received by the firewall from the LAN address at Site B. If replies are not received, move on to the *Troubleshooting IPsec VPNs* section.

If the tunnel was not established initially, it is common for a few pings to be lost during tunnel negotiation, so choosing a higher count or re-running the test is a good practice if the first attempt fails.

Specifying a Ping Source in the Shell

Using the shell on the console or via ssh, the ping command can be run manually and a source address may be specified with the *-S* parameter. Without using - *S* or a static route, the packets generated by ping will not attempt to traverse the tunnel. This is the syntax for a proper test:

```
#  ping -S <Local LAN IP> <Remote LAN IP>
```

Where the *Local LAN IP* is an IP address on an internal interface within in the local subnet definition for the tunnel, and the *Remote LAN IP* is an IP address on the remote router within the remote subnet listed for the tunnel. In most cases this is simply the LAN IP address of the respective pfSense firewalls. Given the site-to-site example above, this is what would be typed to test from the console of the Site A firewall:

```
#  ping -S 10.3.0.1 10.5.0.1
```

If the tunnel is working properly, ping replies will be received by the firewall from the LAN address at Site B. If replies are not received, move on to the *Troubleshooting IPsec VPNs* section.

19.8.11 Client Routing and Gateway Considerations

When the VPN endpoint is the default gateway for a network there are normally no problems with routing. When a client PC sends traffic it will go to its default gateway, over the tunnel, and out the other end. However, if the firewall is not the default gateway for a given network, then other routing measures will need to be taken.

As an example, imagine that the firewall is the gateway at Site B, but not Site A, as illustrated in Figure *Site-to-Site IPsec Where the VPN Endpoint is not the Gateway*. A device, PC1 at Site B, sends a ping to PC2 at Site A. The packet leaves PC1, travels through the firewall at Site B, across the tunnel, out the firewall at Site A, and on to PC2. But what happens on the way back? The gateway on PC2 is a different router entirely. The reply to the ping will be sent to the gateway router and most likely be tossed out, or even worse, it may be sent out the Internet link and be lost that way.

There are several ways around this problem, and any one may be better depending on the circumstances of a given case.

- A static route could be entered into the gateway router that will redirect traffic destined for the far side of the tunnel to the VPN endpoint. Even with this route, additional complexities are introduced because this scenario results in asymmetric routing as covered in *Bypass Firewall Rules for Traffic on Same Interface*.
- A static route could be added to the client systems individually so that they know to send that traffic directly to the VPN endpoint and not via their default gateway. Unless there are only a very small number of hosts that need to access the VPN, this is a management headache and should be avoided.
- In some situations it may be easier to make the VPN endpoint the gateway and let it handle the Internet connection instead of the existing gateway.

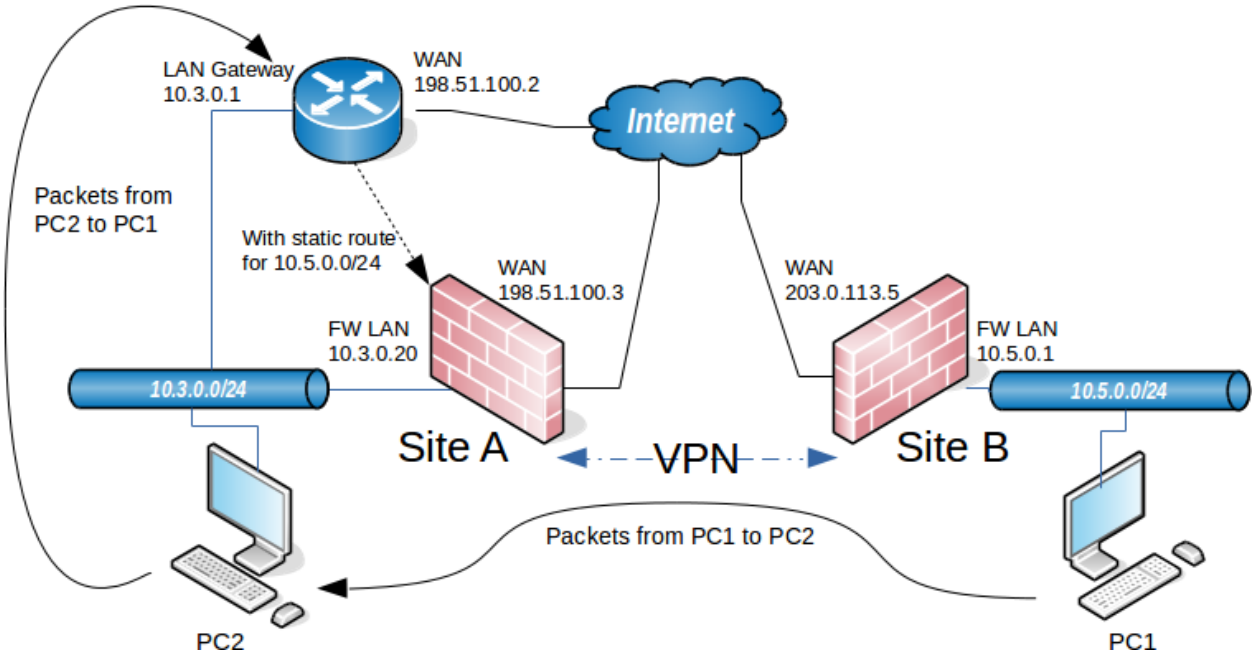


Fig. 4: Site-to-Site IPsec Where the VPN Endpoint is not the Gateway

19.8.12 Configuring Third Party IPsec Devices

Any VPN device which supports standard IPsec may be connected to a device running pfSense® software. pfSense is used in production in combination with numerous vendors' equipment, and will most likely work fine with any IPsec capable devices encountered in other networks. Connecting devices from two different vendors can be troublesome regardless of the vendors involved because of configuration differences between vendors, in some cases bugs in the implementations, and the fact that some of them use proprietary extensions. Some examples are provided at the end of this chapter for several common Cisco devices.

To configure an IPsec tunnel between pfSense and a device from another vendor, the primary concern is to ensure that the phase 1 and 2 parameters match on both sides. For the configuration options on pfSense, where it allows multiple options to be selected, only select one of those options and ensure the other side is set the same. The endpoints will attempt to negotiate a compatible option when multiple options are selected, however that is frequently a source of problems when connecting to third party devices. Configure both ends to what are believed to be matching settings, then save and apply the changes on both sides.

Once the settings match on both ends of the tunnel, attempt to pass traffic over the VPN to trigger its initiation then check the IPsec logs on both ends to review the negotiation. Depending on the situation, the logs from one end may be more useful than those from the opposite end, so it is good to check both and compare. The pfSense side typically provides better information in some scenarios, while on other occasions the other device provides more useful logging. If the negotiation fails, determine whether it was phase 1 or 2 that failed and thoroughly review the settings accordingly, as described in *Troubleshooting IPsec VPNs*. The side that is initiating often cannot see why, so check the logs on the responding side first.

Terminology Differences

Another frequent source of failures is differences in terminology between vendors. Here are a few common things to look out for:

Policy-Based VPN/IPsec The type of IPsec used by pfSense. Policies are defined, such as Phase 2 entries, which control traffic entering the tunnel.

Route-Based VPN/IPsec This style of IPsec is not supported by pfSense, but some vendors or equipment may require it. There is an IPsec interface which routes similar to other interfaces and obeys the routing table, rather than relying on policies.

S2S or L2L Short for Site-to-Site or LAN-to-LAN, distinguished from a mobile client style VPN.

Perfect Forward Secrecy (PFS) Some vendors have different controls for PFS. It may only be a toggle which uses the same value as the Phase 1 **DH Group**, others label it with full text or the acronym, others label it **DH Group**.

Transform Set On Cisco devices, a set of parameters that define Phase 2 handling such as encryption and hash algorithms.

ISAKMP Policy On Cisco devices, a set of parameters that define Phase 1 handling such as authentication, encryption, and hash algorithms, and others.

Proposals On Juniper and Fortigate, sets of options that define parameters for Phase 1 (IKE) or Phase 2 (IPsec) handling.

NAT Exemption or no-nat On Juniper and Cisco, exceptions to NAT that must be made to ensure that traffic traversing a VPN does not have NAT applied.

Lifebytes or Traffic Lifetime Limits on the amount of traffic sent over a VPN before it renegotiates. Not currently supported in the pfSense GUI, if present on a remote device it may need to be disabled.

Encryption Domain or Policy A network definition used in Phase 2 to control which traffic will be handled by IPsec.

Compatible Devices

Any device supporting standard IPsec can be connected with pfSense software. This page lists devices that have been reported to work by users, though it shouldn't be considered complete. Most every device in existence that supports IPsec is in active use somewhere.

- Adtran
- Cisco routers
- Cisco PIX and ASA firewalls
- Checkpoint NG
- DLink VPN Routers
- Draytek VPN routers
- IBM z/OS mainframes
- IPCop
- Juniper routers and firewalls
- Kerio Control
- LANCOM VPN Routers with LCOS
- Linksys VPN Routers
- m0n0wall
- Mikrotik
- Nortel Contivity
- Palo Alto Networks
- Sonicwall
- StoneGate Firewall/VPN
- Ubiquiti Unifi Security Gateway
- Watchguard
- Zyxel firewalls

. . . and many more.

If a device is not listed and is known to work with pfSense software for IPsec, please *submit a documentation update*.

Third Party Firewall Examples

The following examples are for third-party Cisco devices running a hypothetical tunnel to a slightly different example from the one in this chapter. The address details are the same as *IPsec Site-to-Site VPN Example with Pre-Shared Keys* but there are some differences in these examples:

Phase 1 and Phase 2 Encryption 3DES

Phase 1 and Phase 2 Hash SHA1

Phase 1 Lifetime 86400

Cisco PIX OS 6.x

The following configuration is for a Cisco PIX running on 6.x as Site B from the example site-to-site configuration earlier in the chapter.

```
sysopt connection permit-ipsec
isakmp enable outside

!--- Phase 1
isakmp identity address
isakmp policy 1 encryption 3des
isakmp policy 1 hash sha
isakmp policy 1 group 2
isakmp policy 1 lifetime 86400
isakmp policy 1 authentication pre-share
isakmp key aBc123%XyZ9$7qwErty99 address 198.51.100.3 netmask 255.255.255.255 no-
↪xauth no-config-mode

!--- Phase 2
crypto ipsec transform-set 3dessha1 esp-3des esp-sha-hmac
access-list PFSVPN permit ip 10.5.0.0 255.255.255.0 10.3.0.0 255.255.255.0
crypto map dyn-map 10 ipsec-isakmp
crypto map dyn-map 10 match address PFSVPN
crypto map dyn-map 10 set peer 198.51.100.3
crypto map dyn-map 10 set transform-set 3dessha1
crypto map dyn-map 10 set security-association lifetime seconds 3600
crypto map dyn-map interface outside

!--- no-nat to ensure it routes via the tunnel
access-list nonat permit ip 10.5.0.0 255.255.255.0 10.3.0.0 255.255.255.0
nat (inside) 0 access-list nonat
```

Cisco PIX OS 7.x, 8.x, and ASA

Configuration on newer revisions of the PIX OS and for ASA devices is similar to that of the older ones, but has some significant differences. The following example would be for using a PIX running OS version 7.x or 8.x, or an ASA device, as Site B in the site-to-site example earlier in this chapter.

```
crypto isakmp enable outside

!--- Phase 1
crypto isakmp policy 10
  authentication pre-share
  encryption 3des
  hash sha
  group 2
  lifetime 86400

tunnel-group 198.51.100.3 type ipsec-l2l
tunnel-group 198.51.100.3 ipsec-attributes pre-shared-key aBc123%XyZ9$7qwErty99

!--- Phase 2
crypto ipsec transform-set 3dessha1 esp-3des esp-sha-hmac
access-list PFSVPN extended permit ip 10.5.0.0 255.255.255.0 10.3.0.0 255.255.255.0
crypto map outside_map 20 match address PFSVPN
```

(continues on next page)

(continued from previous page)

```
crypto map outside_map 20 set peer 198.51.100.3
crypto map outside_map 20 set transform-set 3dessha1
crypto map outside_map interface outside

!--- no-nat to ensure it routes via the tunnel
access-list nonat extended permit ip 10.5.0.0 255.255.255.0 10.3.0.0 255.255.255.0
nat (inside) 0 access-list nonat
```

Cisco IOS Routers

This shows a Cisco IOS-based router as Site B from the example site-to-site configuration earlier in the chapter.

```
!--- Phase 1
crypto isakmp policy 10
  encr 3des
  authentication pre-share
  group 2
crypto isakmp key aBc123%XyZ9$7qwErty99 address 198.51.100.3 no-xauth

!--- Phase 2
access-list 100 permit ip 10.3.0.0 0.0.0.255 10.5.0.0 0.0.0.255
access-list 100 permit ip 10.5.0.0 0.0.0.255 10.3.0.0 0.0.0.255
crypto ipsec transform-set 3DES-SHA esp-3des esp-sha-hmac
crypto map PFSVPN 15 ipsec-isakmp
  set peer 198.51.100.3
  set transform-set 3DES-SHA
  match address 100

!--- Assign the crypto map to the WAN interface
interface FastEthernet0/0
  crypto map PFSVPN

!--- No-Nat so this traffic goes via the tunnel, not the WAN
ip nat inside source route-map NONAT interface FastEthernet0/0 overload
access-list 110 deny   ip 10.5.0.0 0.0.0.255 10.3.0.0 0.0.0.255
access-list 110 permit ip 10.5.0.0 0.0.0.255 any
route-map NONAT permit 10
  match ip address 110
```

19.8.13 Accessing Firewall Services over IPsec VPNs

With an out of the box configuration, it is not possible to query SNMP or several other services on the LAN interface address of a remote firewall running pfSense® software over an IPsec VPN connection.

Fred Wright explained in a post to the m0n0wall mailing list on September 12, 2004 why this is, and it's the same reason here.

> Due to the way IPsec tunnels are kludged into the FreeBSD kernel, any traffic *initiated* by m0n0wall to go through an IPsec tunnel gets the wrong source IP (and typically doesn't go through the tunnel at all as a result). Theoretically this *shouldn't* be an issue for the *server* side of SNMP, but perhaps the server has a bug (well, deficiency, at least) where it doesn't send the response out through a socket bound to the request packet. You can fake it out by adding a bogus static route to the remote end of the tunnel via the m0n0wall's LAN IP (assuming that's within the near-end tunnel range). A good test is to see whether you can ping something at the remote end of the tunnel (e.g. the SNMP remote) *from* the

m0n0wall. There's an annoying but mostly harmless side-effect to this - every LAN packet to the tunnel elicits a no-change ICMP Redirect.

Most notably this is a problem for UDP services. UDP services reply using the "closest" address to the client as seen from the perspective of the system routing table. Without a route present, that ends up being the IP address of the default gateway on WAN.

Service Binding Workaround

Depending on the service, it may be possible to change the interface binding so that it only listens on the LAN IP address (or the IP address of the internal network on the local end of the VPN) on the firewall. The interface binding for SNMP, NTP, the DNS Forwarder, and several other services can be set in this way.

Static Route Workaround

If changing the service binding is not possible, or for full connectivity between the endpoints, use static routes to work around the situation.

Note: When both endpoints must be accessed, a static route is required on each endpoint.

Setting up a static route is done by first adding a gateway pointing to the LAN IP address of the firewall (See *Gateways*), and then adding a static route using this gateway (See *Static Routes*).

Add Gateway

First, add a gateway for the address of the firewall itself:

- Navigate to **System > Routing** on the **Gateways** tab
- Click  **Add** to add a gateway
- Configure the following settings:

  Interface LAN

  Name `IPsecGW` or another appropriate name, as desired.

  Gateway Enter the LAN IP address of **this** firewall.

  Disable Gateway Monitoring *Checked*

- Click **Save**
- Click **Apply Changes**

Add Route

Now add this route in the GUI:

- Navigate to the **Static Routes** tab
- Click  **Add**
- Configure the following settings:

  Destination Network The **remote** VPN network

  Gateway **IPsecGW** or whichever name was used when creating the gateway

  Description `Allow firewall itself to communicate over VPN`
- Click **Save**
- Click **Apply Changes**

Test

To perform a quick test with *ping using the GUI*:

- Navigate to **Diagnostics > Ping**
- Fill in the following options:

  Hostname `10.0.0.1`

  Source Address *LAN*
- Click  **Ping** to start the test
- Wait for the GUI to display the test results

To use `ping` from the console or ssh, adjust the ping source to enable traffic to traverse the tunnel like so:

```
ping -S <LAN ip> <remote IP address>
```

If the firewall LAN address is `192.168.1.1`, and that IP address is a part of the subnet defined for the IPsec tunnel, to ping `10.0.0.1` on the other side, do this:

```
ping -S 192.168.1.1 10.0.0.1
```

19.8.14 IPsec and IPv6

IPsec is capable of connecting to a tunnel over IPv4 or IPv6 phase 1 peer addresses, but with IKEv1 the tunnel can only contain the same type of traffic inside the tunnel phase 2 definition that is used to pass the traffic outside the tunnel. This means that although either IPv4 or IPv6 may be carried inside of the tunnel, to use IPv6 traffic inside the tunnel it must be connected between IPv6 peer IP addresses, not IPv4. In other words, the inner and outer address family must match, they cannot be mixed.

As with most other shortcomings of IKEv1, this has been addressed in IKEv2. Tunnels using IKEv2 may carry both types of traffic no matter which protocol is used to establish the outer tunnel. With IKEv2, mobile clients may also use both IPv4 and IPv6, provided the client supports it.

See also:

- *IPsec Logs*
- *IPsec Status*
- *IPsec Site-to-Site VPN Example with Pre-Shared Keys*
- *IPsec Site-to-Site VPN Example with Certificate Authentication*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-RADIUS*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-TLS*
- *IPsec Remote Access VPN Example Using IKEv1 with Xauth*
- *IPsec Remote Access VPN Example Using IKEv1 with Pre-Shared Keys*
- *Routing Internet Traffic Through a Site-to-Site IPsec Tunnel*
- *Connecting to Cisco IOS Devices with IPsec*
- *Connecting to Cisco PIX/ASA Devices with IPsec*
- *Connecting to L2TP/IPsec from Android*
- *L2TP/IPsec Remote Access VPN Configuration Example*
- *Troubleshooting IPsec VPNs*

IPsec provides a standards-based VPN implementation that is compatible with a wide range of clients for mobile connectivity, and other firewalls and routers for site-to-site connectivity. It supports numerous third party devices and is being used in production with devices ranging from consumer grade Linksys routers all the way up to IBM z/OS mainframes, and everything imaginable in between. This chapter describes the configuration options available, and how to configure various common scenarios.

See also:

For general discussion of the various types of VPNs available in pfSense® software and their pros and cons, see *Virtual Private Networks*.

pfSense software supports IPsec with IKEv1 and IKEv2, multiple phase 2 definitions for each tunnel, as well as NAT traversal, NAT on Phase 2 definitions, a large number of encryption and hash options, and many more options for mobile clients, including xauth and EAP.

19.8.15 IPsec Terminology

Before delving too deeply into configuration, there are a few terms that are used throughout the chapter that require explanation. Other terms are explained in more detail upon their use in configuration options.

IKE

IKE stands for *Internet Key Exchange*, and comes in two different varieties: IKEv1 and IKEv2. Nearly all devices that support IPsec use IKEv1. A growing number of devices also support the newer IKEv2 protocol which is an updated version of IKE that solves some of the difficulties present in the earlier version. For example, IKEv2 has MOBIKE, which is a standard for mobile clients that allows them to switch addresses dynamically. It also has built-in NAT traversal, and standard mechanisms for reliability similar to DPD. In general IKEv2 provides a more stable and reliable experience, provided both ends support it sufficiently.

ISAKMP Security Association

ISAKMP stands for Internet Security Association and Key Management Protocol. It gives both parties a mechanism by which they can set up a secure communications channel, including exchanging keys and providing authentication.

An ISAKMP Security Association (ISAKMP SA) is a one-way policy which defines how traffic will be encrypted and handled. Each active IPsec tunnel will have two security associations, one for each direction. The ISAKMP Security Associations are setup between the public IP addresses for each endpoint. Knowledge of these active security associations is kept in the Security Association Database (SAD).

Security Policy

A Security Policy manages the complete specifications of the IPsec tunnel. As with Security Associations, these are one-way, so for each tunnel there will be one in each direction. These entries are kept in the Security Policy Database (SPD). The SPD is populated with two entries for each tunnel connection as soon as a tunnel is added. By contrast, SAD entries only exist upon successful negotiation of the connection.

In pfSense software, Security Policies control which traffic will be intercepted by the kernel for delivery via IPsec.

Phase 1

There are two phases of negotiation for an IPsec tunnel. During phase 1, the two endpoints of a tunnel setup a secure channel between using ISAKMP to negotiate the SA entries and exchange keys. This also includes authentication, checking identifiers, and checking the pre-shared keys (PSK) or certificates. When phase 1 is complete the two ends can exchange information securely, but have not yet decided what traffic will traverse the tunnel or its encryption.

Phase 2

In phase 2, the two endpoints negotiate how to encrypt and send the data for the private hosts based on Security Policies. This part builds the tunnel used for transferring data between the endpoints and clients whose traffic is handled by those endpoints. If the policies on both side agree and phase 2 is successfully established, the tunnel will be up and ready for use for traffic matching the phase 2 definitions.

19.9 WireGuard

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.1 WireGuard Settings

WireGuard Tunnel Settings

When creating or editing a WireGuard tunnel, the following options are available:

Enabled Controls whether or not this WireGuard instance is enabled or disabled.

> **Note:** A WireGuard instance cannot be disabled while assigned as an interface.

Description A short text description of this WireGuard instance.

Address A comma-separated list of IPv4 and/or IPv6 addresses which will be assigned to this WireGuard interface.

> **Note:** Use a subnet mask of sufficient size to contain all peers.

Listen Port The local port upon which this WireGuard instance will listen for incoming traffic from peers, and the port from which it will source outgoing packets. The default port is `51820`, additional tunnels must use a different port. The GUI will automatically suggest the next highest available port.

Interface Keys The private and public key pair for this WireGuard instance. The public key is derived from the private key and does not need to be entered separately. The GUI will display the public key automatically when possible. When entering a new private key manually, the public key will be available after saving the tunnel.

The private key will stay only on this firewall, the public key will be copied to peers.

A new set of keys can be generated by the  **Generate** button.

> **Tip:** Click **Copy** under the public key to copy it to the clipboard.

WireGuard Peer Settings

WireGuard peers are defined inside a tunnel entry. They control which remote hosts are allowed to connect to the VPN and how the firewall communicates with these peers.

When creating or editing a WireGuard tunnel, the following options are available:

Description A short text description of this peer.

Endpoint The IP address or hostname of the remote WireGuard peer, from which the peer will connect to this firewall, and to which this WireGuard instance will send traffic destined for this peer.

This can be left empty if the peer endpoint is unknown, such as for dynamic remote access clients. When empty, the tunnel will track the endpoint dynamically based on the key used by the peer. Additionally, when empty, this firewall cannot initiate traffic on the tunnel to the peer until the remote peer sends traffic.

Endpoint Port The port used by the peer for WireGuard traffic. The default port is `51820` if left empty.

> **Note:** If the **Endpoint** is empty, this value is ignored.

Keep Alive An interval, in seconds, at which an empty packet is sent to the peer to keep the session active. This can improve handling through stateful firewalls. Disabled by default.

Public Key The public key of this peer.

Allowed IPs List of networks **on the peer side** which the firewall can reach through this peer. For example, on a site-to-site VPN this would be the tunnel address of the peer and the *peer* LAN subnet.

When tunnel has multiple peers this list allows WireGuard to determine which peer will receive traffic for destinations routed through the WireGuard interface.

The networks listed here also get setup for routing at the operating system level.

> **Warning:** These networks cannot be duplicated between multiple peers, they must be unique.

> **Note:** All traffic may be associated with a peer by using `0.0.0.0/0` for IPv4 or `::/0` for IPv6, but this won't work for a tunnel with multiple peers. Automatic operating system routes are not added for these default route style targets.

> **Tip:** For those familiar with OpenVPN, the internal routing used by WireGuard is similar to `iroute` statements which associate remote networks with specific clients.

Peer WireGuard Address If present, this may be used by functionality which requires knowledge of the WireGuard tunnel address of the peer. For example, when assigning a WireGuard tunnel as an interface (*Assign a WireGuard Interface*), this value can be selected for use as the automatic interface gateway.

> **Note:** This does not affect the underlying WireGuard peer configuration.

This is necessary since the **Allowed IPs** list may not necessarily contain the specific remote peer address, and features such as automatic gateways and configuration export may need to know this address.

Pre-Shared Key An optional Pre-Shared key which provides and additional layer of symmetric-key cryptography on top of the public key cryptography for post-quantum resistance.

> **Warning:** WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.
>
> If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*
>
> WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.2 Design Considerations

One of the main considerations when choosing a WireGuard implementation layout is whether to use one tunnel with many peers, or one tunnel per peer.

Routing to WireGuard Peers

WireGuard uses what it calls "Cryptokey Routing" to map traffic inside WireGuard to a specific peer which is then encrypted using the public key for that peer. In practice, this means that when multiple peers are defined on a WireGuard instance, it must have all networks which will be routed to each peer defined on the peer. This can make managing networks and routes cumbersome.

When there is only one peer on a wireguard interface, it can instead assume that the one peer is the correct destination for all traffic which crosses the interface (e.g. **Allowed IPs** set to `0.0.0.0/0`). And in that case, a routing protocol such as BGP or OSPF can manage the operating system routing to the neighbor instead of static routes.

Design Style

WireGuard does not have a concept of "Client" and "Server" per se, but depending on the configuration the firewall can behave in a manner similar to a "Client" (initiates locally, remote never initiates) or "server" (never initates, remotes always initiate).

Technically every WireGuard tunnel is a peer to peer connection, but there are three main ways a WireGuard tunnel can be configured depending on whether or not a peer endpoint is known or defined:

- Site-to-Site (peer endpoint filled in on both sides)
- Remote Access "Server" (endpoint only filled in on remote peers)
- Remote Access "Client" (endpoint only filled in locally, not on the "server" peer)

Any of those roles can technically be configured no matter how the peer endpoint settings are defined, but not defining an endpoint on one side or the other limits the capacity in which a peer can operate.

Typically, a tunnel is defined with a known peer IP address or hostname for the endpoint, which could take advantage of Dynamic DNS for dynamic peers. This is the most secure method as it locks the tunnel down to specific known peers, but that is not always practical.

In the case of remote access style setups, the peer endpoint address is typically unknown and can change at any time. In this case, the peer endpoint can be left blank and WireGuard will accept connections from any remote address, validating the key instead.

Note: WireGuard supports roaming automatically, and can detect when a peer has changed IP addresses. WireGuard will recognize that authenticated data is coming from a new address and update itself accordingly.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.3 WireGuard Limitations

There are a few limitations of the WireGuard implementation in FreeBSD which must be taken into consideration when deciding if WireGuard fits the needs of a use case.

Response Sourcing

WireGuard does not bind itself to an interface or a specific address on the firewall, but instead can accept traffic on any local IP address. This makes it very flexible, but can cause problems with functionality which requires traffic to use a specific address. When a WireGuard peer contacts the firewall, the firewall will respond from the address the peer used to contact it, if possible.

There are certain cases where this may not function as expected, such as when a peer was communicating with a CARP VIP which changed status.

Remote Peer Endpoint Requirements

When this firewall initiates a packet to a remote WireGuard endpoint, the source will be based on the main IP address on the WAN interface with the default gateway. If a remote endpoint expects to communicate with a different address, it is unlikely to work.

In this case, the problem can be worked around by updating the remote peer to use the correct endpoint address.

If this firewall is the only peer to initiate traffic in the tunnel, an outbound NAT rule could be crafted which matches the incorrect source address and translates it to the expected address.

Another alternative is to remove the remote endpoint from the peer configuration on this firewall, and set the remote endpoint to use the expected peer address for this firewall. If the remote peer always initiates traffic and sends to the expected address, replies will use that address.

High Availability

The address selection behavior mentioned earlier can cause some unexpected behavior with CARP.

For example, if a CARP node goes into maintenance mode during ongoing communication with a peer, the CARP VIP will no longer be available for use by WireGuard and the CARP node may respond to the client using its interface address instead. This can cause the remote peer to continue attempting communication with that specific node, and not the node holding MASTER status for the CARP VIP. This isn't a problem when a CARP node is powered off or reboots, as in those cases the CARP node cannot issue the unexpected responses.

When initiating traffic to a remote endpoint it can't be properly sourced from a CARP VIP. Outbound NAT cannot work around this limitation as the state created by outbound NAT is tied to a specific HA node. If that node fails, the other node cannot utilize the same state as it doesn't match its own local address. Allowing the remote peer to always initiate can work around this limitation.

Multi-WAN

If a client contacts the firewall via its WAN2 address, the firewall will respond from its WAN2 address as expected. However, if this firewall initiates, the traffic will always leave via the interface with the default route unless the routing table sends the traffic by another path.

If the only concern is connectivity, this may be acceptable. However, it is not acceptable if the intent is to cause WireGuard to use WAN2 for its initiated traffic.

For remote peers with static addresses, this can be worked around by adding a static route for the remote endpoint address using the correct WAN gateway.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.4 Configure a WireGuard Tunnel

To configure a WireGuard Tunnel:

- Navigate to **VPN > WireGuard**
- Click  **Add Tunnel**
- Fill in the WireGuard Tunnel settings as described in *WireGuard Tunnel Settings*
- Click  **Add Peer**
- Fill in the WireGuard Peer settings as described in *WireGuard Peer Settings*
- Repeat the add/configure steps for peers if there are multiple peers
- Click **Save**
- Add firewall rules on **Firewall > Rules**, **WAN** tab to allow UDP traffic to the port for this WireGuard tunnel (*WireGuard and Rules / NAT*)
- Add firewall rules on the common **Firewall > Rules**, **WireGuard** tab to pass traffic inside the VPN (*WireGuard and Rules / NAT*)

After configuring the WireGuard instance, there are a few more optional steps depending on the requirements of the use case:

- Navigate to **System > Routing**
- Set the **Default gateway** options to a specific gateway or group, as long as they are not left at *Automatic*

Warning: If the default gateway remains set to *Automatic* the firewall may end up using the WireGuard interface as the default gateway, which is unlikely to be the desired outcome.

- Assign the WireGuard interface as a new OPTx interface (*Assign a WireGuard Interface*)
- Add firewall rules specific to this tunnel on **Firewall > Rules**, **OPTx** tab to pass traffic inside the VPN (*WireGuard and Rules / NAT*)
- Setup one of the alternate routing methods as described in *WireGuard Routing*, if needed.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.5 Assign a WireGuard Interface

Some functionality for WireGuard interfaces depends upon them being assigned as their own interfaces on the firewall. Benefits of assignment include:

- Adds a firewall tab under **Firewall > Rules**
- Allows the interface to be selected for use with NAT rules
- Allows the interface to be selected throughout the GUI and packages for various purposes
- Adds an automatic dynamic gateway for use with routing, policy routing, gateway groups, etc.
- Rules on assigned interface tabs get `reply-to` which ensures return routing will exit back the expected interface for inbound connections.

Automatic WireGuard Gateway Behavior

By default the automatic gateway for an assigned WireGuard interface uses the address of the WireGuard interface itself to nudge traffic into the tunnel. If the **Peer WireGuard Address** value is defined on a peer, and it lies within the WireGuard **Address** subnet, that value will be used for the gateway instead.

In the case of single peer tunnels, using the peer address for the gateway makes more sense as it enables monitoring end-to-end connectivity of the tunnel. It does not make as much sense for tunnels with multiple peers as no one single peer will reflect the connectivity of the tunnel as a whole.

Note: Manual gateways can be defined as needed in addition to the automatic gateway.

Assignment Procedure

To assign the interface:

- Navigate to **Interfaces > Assignments**
- Select the appropriate `wg<number>` interface in the **Available network ports** list

  The description of the tunnel is printed next to the interface name in the list.

- Click  **Add** to assign the interface as a new OPT interface (e.g. OPT1)
- Navigate to the Interface configuration page, **Interfaces > OPTx**
- Check **Enable**
- Enter an appropriate **Description** which will become the interface name (e.g. `WG_S2S`)
- Click **Save**
- Click **Apply Changes**

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.6 WireGuard and Rules / NAT

There are multiple concerns with firewall rules for WireGuard.

External Traffic

Firewall rules must pass traffic on WAN to the WireGuard **Listen Port** for a tunnel if remote WireGuard peers will initiate connections to this firewall. The protocol is always UDP, and the default port is `51820`.

Tunneled Traffic

Firewall rules must pass traffic on WireGuard interfaces to allow traffic inside the VPN, assuming remote connections should be allowed to local internal hosts. Use rules on the WireGuard group tab or rule tabs for assigned interfaces.

Rules on the WireGuard group tab are considered first and can match traffic on any WireGuard interfaces whether or not they are assigned.

Assigned WireGuard interfaces get their own individual rule tabs and will only match traffic on that specific tunnel interface. Rules on assigned WireGuard interface tabs also get `reply-to` which ensures that traffic entering a specific assigned WireGuard interface exits back out the same interface. Without that, return traffic will follow the default gateway.

Warning: Rules on the WireGuard group tab are matched first, so ensure rules on the group tab are removed, disabled, or do not match traffic which requires `reply-to`.

NAT functions on WireGuard interfaces once assigned. Outbound NAT, 1:1 NAT, and port forwards all work as expected.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

19.9.7 WireGuard Routing

WireGuard can work with both static and dynamic routing, depending on the environment.

Static Routing

Several components go into static routing for WireGuard peer networks.

Automatic

The GUI has a mechanism to automatically setup routes to a WireGuard peer. Networks listed in the **Allowed IPs** field will automatically get routing table entries which point the listed networks to the proper WireGuard tunnel interface. The exception to that is default style routes (e.g. `0.0.0.0/0`) which do not get automatic route entries.

Manual

WireGuard routing can be handled manually as well if there is only one peer per tunnel. To setup routes:

Warning: Before assigning the interface, make sure default gateway for the firewall is not set to *Automatic* or the firewall may end up using the `wg` interface as the default gateway, which is unlikely to be the desired outcome.

- Assign the WireGuard interface for this peer (*Assign a WireGuard Interface*)
- Add static routes using the dynamic gateway for the WireGuard tunnel

Dynamic Routing

WireGuard can work with dynamic routing, but there are some special considerations to take into account.

Note: This has only been tested with FRR.

The primary requirement to use dynamic routing with WireGuard is that there can only be one peer per WireGuard tunnel. When more than one peer is connected to a single WireGuard tunnel, WireGuard requires **Allowed IPs** to decide where to send specific networks. In that case, having to define these networks manually negates the purpose of dynamic routing. Using a single peer allows WireGuard to send any traffic it needs across the interface, including arbitrary networks.

BGP BGP works without any special configuration. Define the neighbor using the WireGuard interface address of the peer.

OSPF OSPF works, but needs special settings because it cannot utilize multicast traffic to find neighbors.

In the OSPF settings of FRR:

- Set the WireGuard interface **Network Type** to *Non-Broadcast* mode
- Add a manual entry on the **Neighbors** tab using the WireGuard interface address of the peer

Other routing protocols have not been tested. If a routing protocol relies on broadcast or multicast traffic, it is unlikely to work.

Return Routing

When allowing inbound connections from arbitrary remote networks, use rules only on assigned WireGuard interface tabs only to ensure proper return routing.

Assigned WireGuard interfaces get their own individual rule tabs and will only match traffic on that specific tunnel interface. Rules on assigned WireGuard interface tabs also get `reply-to` which ensures that traffic entering a specific assigned WireGuard interface exits back out the same interface. Without that, return traffic will follow the default gateway.

See also:

- *WireGuard Remote Access VPN Configuration Example*
- *WireGuard Site-to-Site VPN Configuration Example*
- *WireGuard VPN Client Configuration Example*

19.9.8 WireGuard Overview

WireGuard is a new VPN Layer 3 protocol designed for speed and simplicity. It performs nearly as fast as hardware-accelerated IPsec and has only a small number of options in its configuration.

Due to this simplicity, WireGuard lacks many of the conveniences of more complicated VPN types which can help automate large deployments. Thus, while its performance scales well, the management can become cumbersome for large numbers of peers.

WireGuard behaves unlike other traditional VPN types in several ways:

- It operates completely in the kernel
- Configuration is placed directly on the interfaces
- It has no concept of connections or sessions
- It has no facilities for user authentication
- There is no service daemon to stop or start
- There is minimal logging from the kernel
- It does not bind to a specific interface or address on the firewall, it accepts traffic to any address on the firewall on its specified port

WireGuard instances consist of a tunnel and one or more peer definitions which contain of the necessary keys and other configuration data.

WireGuard interfaces carry Layer 3 information and above.

Before WireGuard can be used, upgrade to the latest version of pfSense Plus or pfSense CE and install the experimental WireGuard package from the *Package Manager*.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

See also:

- *L2TP Server Configuration*
- *Troubleshooting Cisco VPN Pass Through*

VPNs provide a means of tunneling traffic through an encrypted connection, preventing it from being seen or modified in transit. pfSense® software offers several VPN options: IPsec, OpenVPN, WireGuard and L2TP. This section provides an overview of VPN usage, the pros and cons of each type of VPN, and how to decide which is the best fit for a particular environment. Subsequent sections discuss each VPN option in detail.

L2TP is purely a tunneling protocol and does not offer any encryption of its own. It is typically combined with another method of encryption such as IPsec in transport mode. Because of this, it doesn't fit in with most of the discussion in this chapter. See *L2TP VPN* for more information on L2TP.

19.10 PPTP Warning

pfSense software does not include a PPTP server. Despite the attraction of its convenience, PPTP **must not be used** under any circumstances because it is no longer secure. This is not specific to the implementation of PPTP that was in pfSense software; Any device that utilizes PPTP is no longer secure.

PPTP relies upon MS-CHAPv2 which has been completely compromised. Intercepted traffic can be decrypted by a third party 100% of the time, so consider any traffic carried in PPTP unencrypted. Migrate to another VPN type as soon as possible. More information on the PPTP security compromise can be found at https://isc.sans.edu/diary/End+of+Days+for+MS-CHAPv2/13807 and https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2/.

CHAPTER

TWENTY

L2TP VPN

20.1 L2TP and Firewall Rules

By default, when the L2TP server is enabled, firewall rules **will not** be automatically added to the chosen interface to permit UDP port *1701*. A firewall rule must be added to whichever interface the L2TP traffic will be entering, typically WAN, the WAN containing the default gateway, or IPsec.

20.2 L2TP and Multi-WAN

L2TP uses UDP port 1701. Because L2TP relies on UDP, the server may have issues using any WAN that is not the default gateway. The daemon will respond from the firewall using the closest address to the client, following the routing table, which is the WAN with the default gateway for remote clients.

20.3 L2TP Server Configuration

To use L2TP, first browse to **VPN > L2TP**. Select **Enable L2TP server**.

> **Warning:** L2TP is **not** a secure protocol by itself; it only provides tunneling, **it does not perform encryption**.

See also:

L2TP/IPsec is a way to secure L2TP traffic by sending it through an encrypted IPsec tunnel. This may be used in combination with a mobile IPsec setup to configure L2TP+IPsec; see *L2TP/IPsec Remote Access VPN Configuration Example* for details.

20.3.1 Interface

The **Interface** setting controls where the L2TP daemon will bind and listen for connections. This is typically the *WAN* interface accepting inbound connections.

20.3.2 IP Addressing

Before starting, determine which IP addresses to use for the L2TP server and clients and now many concurrent clients to support.

Server Address An *unused* IP address outside of the **Remote Address Range**, such as 10.3.177.1 as shown in Figure *L2TP IP Addressing*.

Remote Address Range Usually a new and unused subnet, such as 10.3.177.128/25 (.128 through .255). These are the addresses to be assigned to clients when they connect.

Number of L2TP users Controls how many L2TP users will be allowed to connect at the same time, in this example *13* has been selected.

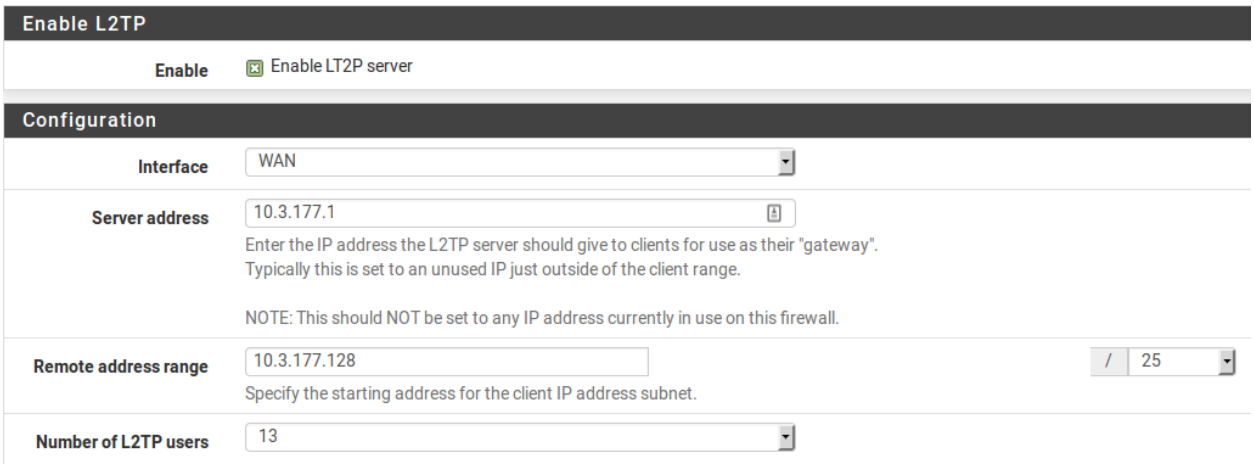


Fig. 1: L2TP IP Addressing

DNS servers can also be defined for end users when needed. Fill in the **Primary** and **Secondary L2TP DNS server** fields with the DNS server IP addresses for connecting clients.

20.3.3 Authentication

Secret Required by some L2TP implementations, similar to a group password or pre-shared key. Support for this varies from client to client. Leave the field blank unless it is known to be required. If required, enter and confirm the secret.

Authentication Type Decides between *PAP*, *CHAP*, or *MS-CHAPv2* authentication for users. Support for this can vary from client to client and it may also depend on the RADIUS server as well. The *CHAP* based types are more secure, but *PAP* is more widely compatible.

Users may be authenticated from the local user database, or via an external RADIUS server. This can be used to authenticate L2TP users from Microsoft Active Directory (see *Authenticating from Active Directory using RADIUS/NPS*) as well as numerous other RADIUS capable servers.

If using RADIUS, check the **Use a RADIUS server for authentication** box and fill in the RADIUS server and shared secret. For authentication using the local user database, leave that box unchecked. Users must be added manually on the **Users** tab of the **VPN > L2TP** screen unless using RADIUS. See *Adding Users* below for more details on the built-in authentication system.

20.3.4 Save changes to start L2TP server

After filling in the aforementioned items, click **Save**. This will save the configuration and launch the L2TP server.

20.3.5 Configure firewall rules for L2TP clients

Browse to **Firewall > Rules** and click the **L2TP VPN** tab. These rules control traffic from L2TP clients. Until a firewall rule has been added to allow traffic, all traffic initiated from connected L2TP clients will be blocked. Traffic initiated from the LAN to L2TP clients is controlled using LAN firewall rules. Initially an allow all rule may be desired here for testing purposes as shown in Figure *L2TP VPN Firewall Rule*, and once functionality has been verified, restrict the ruleset as desired.

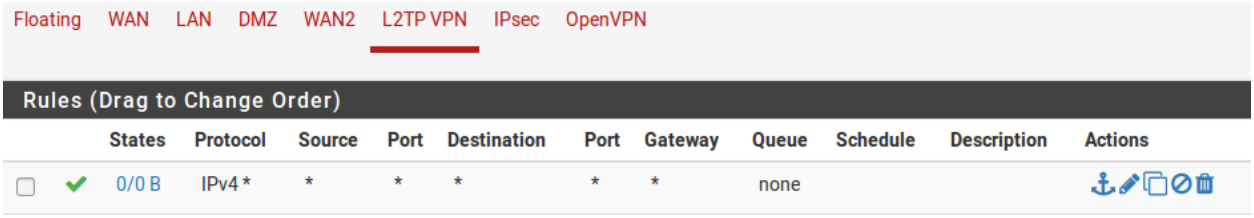


Fig. 2: L2TP VPN Firewall Rule

Note: Remember that a rule must also be added to the interface receiving the L2TP traffic, typically WAN or IPsec, to pass UDP to the firewall with a destination port of `1701`.

20.3.6 Adding Users

Adding users to the built-in L2TP users system is simple. To add local users:

- Navigate to **VPN > L2TP**, **Users** tab. The users screen as shown in Figure *L2TP Users Tab* will be presented.
- Click  **Add** to show the form used to add users.



Fig. 3: L2TP Users Tab

- Enter the **Username**, **Password** and **Confirm Password** for a user, as in Figure *Adding a L2TP User*.
- Enter a static **IP assignment** if desired.
- Click **Save**, and then the user list will return.
- Repeat the process for each user to add.

To edit an existing user, click  . Users may be deleted by clicking  .

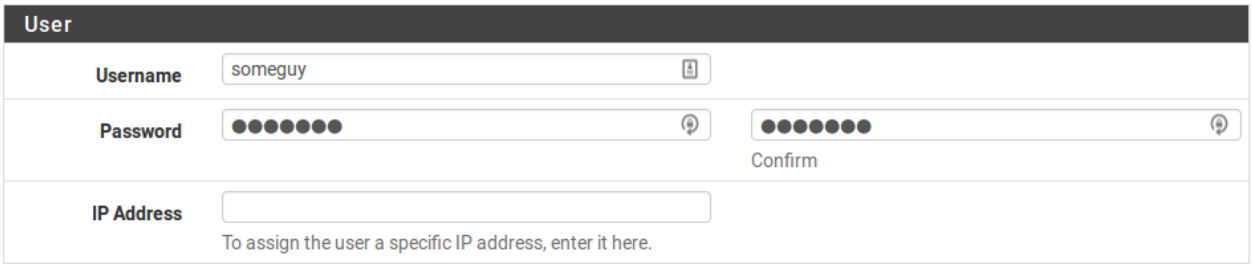


Fig. 4: Adding a L2TP User

See also:

- *L2TP/IPsec Remote Access VPN Configuration Example*
- *Troubleshooting L2TP*
- *L2TP Logs*

pfSense® software can act as an L2TP VPN server. L2TP is purely a tunneling protocol that offers no encryption of its own, so it is typically combined with some other encryption technique, such as IPsec.

Warning: pfSense supports L2TP/IPsec, however, some clients will not work properly in many common scenarios. The most common problem scenario is Windows clients behind NAT, in that case the Windows client and the strongSwan IPsec daemon are not fully compatible, which leads to failure. In these situations, we recommend using IKEv2 instead.

See also:

IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2 contains a walkthrough for configuring IKEv2, which is a much more flexible solution.

See also:

For general discussion of the various types of VPN implementations available in pfSense and their pros and cons, see *Virtual Private Networks*.

20.4 L2TP Security Warning

L2TP on its own is not encrypted, so it is not intended for private traffic. Some devices, such as Android, offer an L2TP-only client which is capable of connecting back to pfSense but it should only be used for traffic that is already encrypted, or if the traffic is not considered private. For example, tunneling Internet traffic so it appears to originate from another location.

CHAPTER

TWENTYONE

SERVICES

21.1 DHCPv4 Server

To alter the behavior of the IPv4 DHCP server, navigate to **Services > DHCP Server** in the web interface. The behavior of the IPv4 DHCP server is controlled there, along with static IP address mappings and related options such as static ARP.

21.1.1 Choosing an Interface

The DHCP configuration page contains a tab for each interface with a static IP address. Each interface has its own separate DHCP server configuration, and they may be enabled or disabled independently of one another. Before making any changes, visit the tab for the correct interface.

21.1.2 General Options

Enable The first setting on the tab enables or disables DHCP service for the interface. To turn on DHCP for the interface, check **Enable DHCP server on [name] interface**. To disable the service, uncheck the box instead.

Deny unknown clients Under normal circumstances, the DHCP server will answer requests from any client requesting a lease. In most environments this is normal and acceptable behavior, but in restricted or secure environments this behavior is undesirable. With this option set, only clients with static mappings defined will receive leases. This is a more secure practice but is much less convenient. This option is per-pool, meaning that if unknown clients are denied in the default range, another pool of IP addresses may be defined that does not have the setting checked. The DHCP server will assign clients IP addresses out of that alternate pool instead.

Note: This will protect against low-knowledge users and people who casually plug in devices. Be aware, however, that a user with knowledge of the network could hardcode an IP address, subnet mask, gateway, and *DNS* which will still give them access. They could also alter/spoof their MAC address to match a valid client and still obtain a lease. Where possible, couple this setting with static ARP entries, access control in a switch that will limit MAC addresses to certain switch ports for increased security, and turn off or disable unused switch ports.

Subnet The network address of the interface subnet, for reference purposes.

Subnet Mask The subnet mask for the interface subnet, for reference purposes.

Available Range The range of available addresses inside the interface subnet, for reference and to help determine the desired range for DHCP clients. The network address and broadcast address are excluded, but interface addresses and Virtual IP addresses are not excluded.

Range This defines the DHCP address range, also referred to as the Scope or Pool. The two boxes for **Range** tell the firewall the first and last address for use as a DHCP pool. Addresses between the entered values, inclusive, will be used for clients which request addresses via DHCP. The range must be entered with the lower number first, followed by the higher number. For example, the default LAN DHCP range is based off of the subnet for the default LAN IP address. It is `192.168.1.100` to `192.168.1.199`. This range can be as large or as small as the network needs, but it must be wholly contained within the subnet for the interface being configured.

21.1.3 Additional Pools

The **Additional Pools** section defines extra pools of addresses inside of the same subnet. These pools can be used to craft sets of IP addresses specifically for certain clients, or for overflow from a smaller original pool, or to split up the main pool into smaller chunks with a GAP of non-DHCP IP addresses in the middle of what used to be the pool. A combination of the MAC Address Control options may be used to guide clients from the same manufacturer into a specific pool, such as VoIP phones.

To add a new pool, click  **Add Pool** and the screen will switch to the pool editing view, which is nearly the same as the normal DHCP options, except a few options that are not currently possible in pools are omitted. The options behave the same as the others discussed in this section. Items left blank will, by default, fall through and use the options from the main DHCP range.

Note: See the MAC Address Control section below for specifics on directing clients into or away from pools.

21.1.4 Servers

WINS Servers Two WINS Servers (Windows Internet Name Service) may be defined that will be passed on to clients. If one or more WINS servers is required, enter their IP addresses here. The actual servers do not have to be on this subnet, but be sure that the proper routing and firewall rules are in place to let them be reached by client PCs. If this is left blank, no WINS servers will be sent to the client.

DNS Servers The DNS Servers may or may not need filled in, depending on the firewall configuration. If the built-in DNS Resolver or DNS Forwarder is used to handle DNS, leave these fields blank and pfSense® will automatically assign itself as the DNS server for client PCs. If the DNS forwarder is disabled and these fields are left blank, pfSense will pass on whichever DNS servers are defined under **System > General Setup**. To use custom DNS Servers instead of the automatic choices, fill in the IP addresses for up to four DNS servers here. In networks with Windows servers, especially those employing Active Directory, it is recommended to use those servers for client DNS. When using the DNS Resolver or DNS forwarder in combination with CARP, specify the CARP Virtual IP address on this interface here.

21.1.5 Other Options

Gateway This may also be left blank if this firewall is acting as the gateway for the network on this interface. If that is not the case, fill in the IP address for the gateway to be used by clients on this interface. When using CARP, fill in the CARP Virtual IP address on this interface here.

Domain Name Specifies the domain name passed to the client to form its fully qualified hostname. If the **Domain Name** is left blank, then the domain name of the firewall it sent to the client. Otherwise, the client is sent this value.

Domain Search List Controls the DNS search domains that are provided to the client via DHCP. If multiple domains are present and short hostnames are desired, provide a list of domain names here, separated by a semicolon. Clients will attempt to resolve hostnames by adding the domains, in turn, from this list before trying to find them externally. If left blank, the Domain Name option is used.

Note: The **Domain Search List** is provided via DHCP option 119. As of this writing, no Windows DHCP *client* of any version supports DHCP option 119. Other operating systems such as BSD, Linux, and OS X do support obtaining the Domain Search List via DHCP option 119.

Default lease time Controls how long a lease will last when a client does not request a specific lease length. Specified in seconds, default value is `7200` seconds (2 hours)

Maximum lease time Limits a requested lease length to a stated maximum amount of time. Specified in seconds, default value is 86400 seconds (1 day).

Failover Peer IP If this system is part of a High Availability failover cluster, enter the real IP address of the other system in this subnet here. Do not enter a CARP Virtual IP address.

Note: When **Failover Peer IP** is configured in a High Availability setup, the failover node should be available when the service is started to allow lease pool information to be synchronized; failing this, the DHCPD service will not respond to DHCPDISOVER requests.

Static ARP This checkbox works similar to denying unknown MAC addresses from obtaining leases, but takes it a step further in that it also restricts any unknown MAC address from communicating with this firewall. This stops would-be abusers from hardcoding an unused address on this subnet, circumventing DHCP restrictions.

Note: When using static ARP, all systems that need to communicate with the firewall must be listed in static mappings before activating this option, especially the system being used to connect to the pfSense GUI. Also be aware that this option may prevent people from hardcoding an IP address and talking to the firewall, but it does not prevent them from reaching each other on the local network segment.

Time Format Change By default, the ISC DHCP daemon maintains lease times in UTC. When this option is checked, the times on the DHCP Leases status page are converted to the local time zone defined on the firewall.

Statistics Graphs This option, disabled by default, activates RRD graphing for monitoring the DHCP pool utilization.

21.1.6 Dynamic DNS

For Dynamic DNS settings, click **Display Advanced** to the right of that field, which displays the following options:

Enable Check the box to enable registration of DHCP client names in DNS using an external (non-pfSense) DNS server.

DDNS Domain The domain name used for registering clients in DNS

Primary DDNS Address The DNS server used for registering clients in DNS

DNS Domain Key The encryption key used for DNS registration

DNS Domain Key Secret The secret for the key used for DNS registration

21.1.7 MAC Address Control

For MAC Address Control, click **Display Advanced** to show the lists of allowed and denied client MAC addresses. Each list is comma-separated and contains portions of MAC addresses. For example, a group of VoIP phones from the same manufacturer may all start with the MAC address `aa:bb:cc`. This can be leveraged to give groups of devices or users separate DHCP options.

Allow A list of MAC Addresses to allow in this pool. If a MAC address is in the allow box, then all others will be denied except the MAC address specified in the allow box.

Deny A list of MAC Addresses to deny from this pool. If a MAC address is in the deny list, then all others are allowed.

It is best to use a combination of allow and deny to get the desired result, such as: In the main pool, leave allow blank and deny `aa:bb:cc`. Then in the VoIP pool, allow `aa:bb:cc`. If that extra step is not taken to allow the MAC prefix in the additional pool, then other non-VoIP phone clients could receive IP addresses from that pool, which may lead to undesired behavior.

This behavior may also be used to blacklist certain devices from receiving a DHCP response. For example to prevent Example brand printers from receiving a DHCP address, if MAC addresses all start with `ee:ee:ee`, then place that in the deny list of each pool.

21.1.8 NTP Servers

To specify NTP Servers (Network Time Protocol Servers), click the **Display Advanced** button to the right of that field, and enter IP addresses for up to two NTP servers.

21.1.9 TFTP Server

click the **Display Advanced** button next to **TFTP** to display the TFTP server option. The value in the TFTP Server box, if desired, must be an IP address or hostname of a TFTP server. This is most often used for VoIP phones, and may also be referred to as "option 66" in other documentation for VoIP and DHCP.

21.1.10 LDAP URI

click the **Display Advanced** button next to **LDAP** to display the **LDAP Server URI** option. **LDAP Server URI** will send an LDAP server URI to the client if requested. This may also be referred to as DHCP option 95. It takes the form of a fully qualified LDAP URI, such as `ldap://ldap.example.com/dc=example,dc=com`. This option can help clients using certain kinds of systems, such as OpenDirectory, to find their server.

21.1.11 Additional BOOTP/DHCP Options

Other numeric DHCP options can be sent to clients using the **Additional BOOTP/DHCP Options** controls. To view these options, click **Display Advanced** in this section. To add a new option, click  **Add**.

Number The DHCP option code number. IANA maintains a list of all valid DHCP options.

Type The choices and formats for each type may be a little counter-intuitive, but the labels are used directly from the DHCP daemon. The proper uses and formats are:

Text Free-form text to be sent in reply, such as `http://www.example.com/wpad/wpad.dat` or `Example Company`.

String A string of hexadecimal digits separated by a colon, such as `c0:a8:05:0c`.

Boolean Either `true` or `false`.

Unsigned 8, 16, or 32-bit Integer A positive Integer that will fit within the given data size, such as `86400`.

Signed 8, 16, or 32-bit Integer A positive or negative Integer that will fit within the given data size, such as `-512`.

IP address or host An IP address such as `192.168.1.1` or a hostname such as `www.example.com`.

Value The value associated with this numeric option and type.

For more information on which options take a specific type or format, see the linked list above from the IANA.

Note: When using numbered custom options, be careful of the type. Some will be OK on text/string but others are not.

For example, DHCP options for code 132 (and presumably 133) for VLAN ID must be set for a type of unsigned integer 32.

21.1.12 Network Booting

To view the Network boot settings, click  in the **Network Booting** section header bar.

Enable Check to enable network booting options in DHCP

Next Server The IP address from which boot images are available

Default BIOS file name File name for the boot image (Non-UEFI)

UEFI 32 bit file name File for 32-bit UEFI booting

UEFI 64 bit file name File for 64-bit UEFI booting

Root Path String to target a specific device as the client's root filesystem device, such as `iscsi:(servername):(protocol):(port):(LUN):targetname`.

21.1.13 Save Settings

After making changes, click **Save** before attempting to create static mappings. Changes to settings will be lost if the browser leaves this page without saving.

21.1.14 Static Mappings

Static DHCP mappings express a preference for which IP address will be assigned to a given client based on its MAC address. In a network where unknown clients are denied, this also serves as a list of "known" clients which are allowed to receive leases or have static ARP entries. Static mappings can be added in one of two ways:

- From this screen, click  **Add**.
- Add them from the DHCP leases view, which is covered later in this chapter.

On this screen, only the **MAC address** is necessary.

MAC Address The client MAC address which identifies the host to deliver options on this page, or by entering only the MAC address, it will be added to the list of known clients for use when the **Deny unknown clients** option is set.

Note: Client MAC address can be obtained from a command prompt on most platforms. On UNIX-based or UNIX-work-alike operating systems including Mac OS X, typing `ifconfig -a` will show the MAC address for each interface. On Windows-based platforms, `ipconfig /all` will show the MAC address. The MAC address may also sometimes be found upon a sticker on the network card, or near the network jack for integrated adapters. For hosts on the same subnet, the MAC can be determined by pinging the IP address of the host and then running `arp - a`.

Client Identifier An ID sent by the client to identify itself.

IP Address The IP address field is needed if this will be a static IP address mapping instead of only informing the DHCP server that the client is valid. This IP address is a preference, not a reservation. Assigning an IP address here will not prevent someone else from using the same IP address. If this IP address is in use when this client requests a lease, it will instead receive an address from the general pool. For this reason, the pfSense WebGUI does not allow assigning static IP mappings inside of the DHCP pool.

Hostname The hostname of the client. This does not have to match the actual hostname set on the client. The hostname set here will be used when registering DHCP addresses in the DNS forwarder.

Description Cosmetic only, and available for use to help track any additional information about this entry. It could be the name of the person who uses the PC, its function, the reason it needed a static address, or the administrator who added the entry. It may also be left blank.

ARP Table Static Entry If checked, this entry will receive a static ARP entry in the OS tying this IP address to this MAC address.

Note: If this option is used rather than using the global static ARP option, it does not prevent that MAC address from using other IP addresses, it only prevents other MAC addresses from using this

IP address. In other words, it prevents another machine from using that IP to reach the firewall, but it doesn't stop the user from changing their own IP address to something different.

The remaining options available to set for this client are the same in behavior to the ones found earlier in this section for the main DHCP settings.

Click **Save** to finish editing the static mapping and return to the DHCP Server configuration page.

21.2 DHCPv6 Server

The DHCPv6 server in pfSense® software will hand out addresses to DHCPv6 clients and automatically configure them for network access. By default, the DHCPv6 server is enabled on the LAN interface and set to use a prefix obtained by tracking WAN's DHCPv6 delegation.

The DHCPv6 server page, found under **Services > DHCPv6 Server**, has a tab for each available interface. The DHCPv6 daemon can only run and be configured on interfaces with a Static IP address, so if a tab for an interface is not present, check that it is enabled and set with a Static IP. It is not currently possible to adjust settings for tracked interface DHCP service.

The DHCPv6 server cannot be active on any interface if the *DHCPv6 Relay* service is in use.

21.2.1 DHCP Instance Options

For each Interface, there are many options to choose from. At a minimum, the **Enable** box must be checked on the interface tab and an address range (starting and ending IPv6 addresses) to use for DHCPv6 clients must be defined. For the DHCPv6 server to be active on the network, *Router Advertisements* must also be set to either *Managed* or *Assisted* mode on the **Router Advertisements** tab.

The other settings may be configured, but are optional.

Note: DHCPv6 does not provide gateway information. *Router Advertisements* tell hosts on the network how to reach a router. DHCPv6 is for other host configuration such as DNS, delegation, and so on.

See the *DNS Forwarder* article for information on the default DNS server behavior.

Some other options which may be set for clients include Network booting options, LDAP URI, and the ability to add in any custom DHCP option number and value.

21.2.2 DHCPv6 Range

The **Range** parameter works similarly to the same setting on IPv4 but it is worth mentioning again here due to the differences in IPv6 addressing.

Given the vast amount of space available inside even a /64, a good trick is to craft a range that restricts hosts to use an easy to remember or recognize range. For example, Inside a /64 such as `2001:db8:1:1::`, set the DHCPv6 range be: `2001:db8:1:1::d:0000` to `2001:db8:1:1::d:FFFF`, using the `d` in the second to last section of the address as a sort of shorthand for "DHCP". That example range contains 2^16 (65,536) IPs, which is extremely large by today's IPv4 standards, but only a small portion of the whole /64.

21.2.3 DHCPv6 Prefix Delegation

Prefix delegation, covered earlier in *DHCP6 Prefix Delegation* and *Track Interface*, allows automatically dividing and allocating a block of IPv6 addresses to networks that will live behind other routers and firewall that reside downstream from pfSense (e.g. in the LAN, DMZ, etc). Most users acting in a client capacity will not need this and will likely leave it blank.

Prefix delegation can be used to hand out /64 chunks of a /48 to routers automatically, or any other combination, so long as the range is set on the boundaries of the desired delegation size. The downstream router obtains an IPv6 address and requests a delegation, and the server allocates one and dynamically adds a route so that it is reachable via the assigned DHCPv6 address given to the client.

The **Prefix Delegation Range** Sets the start and end of the delegation pool. The range of IPv6 addresses specified here must be routed to this firewall by upstream routers. For example, to allocate /60 networks to downstream firewalls out of a given range, then one could specify `2001:db8:1111:F000::` to `2001:db8:1111:FFF0::` with a Prefix Delegation Size of *60*. This allocates a /60 (16 subnets of size /64) to each downstream firewall that requests a delegation so that they can in turn use those for their LAN, VPNs, DMZ, etc. Downstream firewalls can even further delegate their own allocation to routers behind them. Note that in this example, 16 delegations would be possible. Adjust the range and size as needed.

When crafting the values for the range and delegation size, keep in mind that the range must start and end on boundaries that align with the desired prefix size. In this /60 example, the range could not start or end on anything that has a value in the places to the right of the second value in the fourth section of the address, so it can start on `2001:db8:1111:F500::` but **not** `2001:db8:1111:F550::`.

21.2.4 DHCPv6 Static Mappings

Static mappings on DHCPv6 work differently than IPv4. On IPv4, the mappings were performed using the MAC address of the PC. For IPv6, the designers decided that wasn't good enough, since the MAC address of a PC could change, but still be the same PC.

Enter, the **DHCP Unique Identifier**, or **DUID**. The DUID of the host is generated by the operating system of the client and, in theory, will remain unique to that specific host until such time as the user forces a new DUID or the operating system is reinstalled. The DUID can range from 12 to 20 bytes, and varies depending on its type.

The **DUID** field on the static mapping page expects a DUID for a client PC in a special format, represented by pairs of hexadecimal digits, separated by colons, such as `00:01:00:01:1b:a6:e7:ab:00:26:18:1a:86:21`.

How to obtain this DUID depends on the operating system. The easiest way is to allow the PC to obtain a lease via DHCPv6, and then add an entry from the DHCPv6 Leases View (Status DHCPv6 Leases). In Windows, it can be found as DHCPv6 Client DUID in the output of `ipconfig /all`.

Note: On Windows, the DUID is generated at install time, so if a base image is used and workstations are cloned from there, they can all end up with the same DUID, and thus all end up pulling the same IPv6 address over DHCPv6.

Clear the DUID from the registry before making an image to clone, by issuing the following command:

```
reg delete HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters /f /v Dhcpv6DUID
```

That command may also be run on a working system to reset its DUID if needed.

DUID Format

The DUID format is listed on the page, but it roughly follows the format:

```
DUID-LLT - ETH -- TIME --- ---- address ----
```

DUID-LLT is link-layer plus time, which means it uses the link type of a network interface on the system (Generally 00:01 to indicate the format, plus 00:01, or 00:06 for Ethernet), plus the timestamp at which the DUID was generated in hex, plus the MAC address of the first NIC. It may be difficult or impossible to predict a system's DUID. Unless the operating system has a way to look it up, it may be best to allow the client to obtain a dynamic lease and then copy the DUID from the leases view.

21.2.5 Numbered Options Notes

When using numbered custom options, be careful of the type. Some will be OK on text/string but others are not. Also beware that numbered options do NOT correspond exactly to the DHCP numbered options for IPv4

For more information on DHCP option numbers and types, see https://tools.ietf.org/html/draft-ietf-dhc-v6opts-00

21.3 IPv6 Router Advertisements

Automatic address assignment for IPv6 works quite a bit differently than IPv4. Even so, most of the DHCP options are similar, but there are notable differences in behavior in how things are assigned and also how items like the gateway are handed off to clients. Unless otherwise noted, options of the same name work the same for DHCP and DHCPv6. DHCPv6 and Router Advertisements (RA) are configured under **Services > DHCPv6 Server/RA**. Under that page there are two tabs: One for **DHCPv6 Server** and one for **Router Advertisements**.

21.3.1 DHCPv6 vs Stateless Address Autoconfiguration

There are a few clients that do not have support for DHCPv6. Some clients only support Stateless Address Autoconfiguration, or SLAAC for short. There is no way for the firewall to have direct knowledge of a list of hosts on the segment using SLAAC addresses, so for some environments it is much less desirable because of the lack of control and reporting of addresses. Consider address tracking and operating system support requirements when deciding how to allocate IPv6 addresses to clients on the network.

Many operating systems such as Windows, OS X, FreeBSD, Linux, and their cousins contain DHCPv6 clients that are capable of obtaining addresses as expected via DHCPv6. Some lightweight or mobile operating systems such as Android do not contain a DHCPv6 client and will only function on a local segment with IPv6 using SLAAC.

21.3.2 Router Advertisements (Or: "Where is the DHCPv6 gateway option")

In IPv6, a router is located through Router Advertisement (RA) messages sent from routers instead of by DHCP; IPv6-enabled routers that support dynamic address assignment are expected to announce themselves on the network to all clients. As such, DHCPv6 does not include any gateway information. So clients can obtain their addresses from DHCPv6 or SLAAC, but unless they are statically configured, they always locate their next hop by using RA packets sent from available gateways.

To enable the RA service:

- Navigate to **Services > DHCPv6 Server/RA**
- Click the interface tab for the interface being configured

- Click the **Router Advertisements** tab
- Select a mode other than *Disabled* from the **Router Mode** drop-down list
- Click **Save**

The other options to control RA behavior may be set as needed for the network:

Router Advertisement Modes The modes for the RA daemon control the services offered by pfSense®, announce the firewall as an IPv6 router on the network, and direct clients on how to obtain addresses.

Disabled The RA daemon is disabled and will not run. IPv6 gateways must be entered manually on any client hosts.

Router Only This firewall will send out RA packets that advertise itself as an IPv6 router. DHCPv6 is disabled in this mode.

Unmanaged The firewall will send out RA packets and clients are directed to assign themselves IP addresses within the interface subnet using SLAAC. DHCPv6 is disabled in this mode.

Managed The firewall will send out RA packets and addresses will **only** be assigned to clients using DHCPv6.

Assisted The firewall will send out RA packets and addresses can be assigned to clients by DHCPv6 or SLAAC.

Stateless DHCP The firewall will send out RA packets and addresses can be assigned to clients by SLAAC while providing additional information such as DNS and NTP from DHCPv6.

Router Priority If multiple IPv6 routers exist on the same network segment, they can indicate to clients in which order they should be used. If a high priority router becomes unavailable, clients will try a normal priority router, and finally a low priority router. Select either *Low*, *Normal*, or *High* from the list. If there is only one router on the network, use *Normal*.

Default Valid Lifetime Length of time, specified in seconds, that the advertised prefix will be valid. The default value is 86400 seconds (one day)

Default Preferred Lifetime Length of time, specified in seconds, that the client addresses generated in this prefix using SLAAC are valid. The default value is 86400 seconds (one day)

RA Subnets This section allows defining a list of subnets for which this firewall will send RA packets. Enter as many subnets as needed, each with an appropriate prefix (typically *64*.). To create an additional row for another subnet, click  **Add**.

DNS Settings Obtaining DNS information from RA messages is not universally supported, but for clients that do support it, using SLAAC to give an IP address and DNS from RA can do away with the need for using DHCPv6 entirely.

DNS Servers Enter up to three IP addresses for DNS Servers, or leave the fields blank to use the system default DNS servers or DNS Resolver/DNS forwarder if enabled.

Domain Search List Operates identically to the DHCP option of the same name.

Use same settings as DHCPv6 server When checked, these values will be pulled from the DHCPv6 options automatically.

21.4 DHCPv4 & DHCPv6 Relay

DHCP requests are broadcast traffic. Broadcast traffic is limited to the broadcast domain where it is initiated. To provide DHCP service on a network segment without a DHCP server, use the DHCP relay to forward those requests to a defined server on another segment.

Warning: It is not possible to run both a DHCP server and a DHCP Relay at the same time. To enable the DHCP relay, first disable the DHCP server on each interface.

To configure the DHCP Relay:

- Disable the DHCP Server on every interface
- Navigate to **Services > DHCP Relay**
- Click the tab for the interface to use with DHCP Relay
- Configure the options as follows:

  Enable DHCP Relay Checked

  Append circuit ID and agent ID to requests Check this to add a circuit ID (pfSense® interface number) and the agent ID to the DHCP request. This may be required by the DHCP server on the other side, and can help distinguish where the requests originated.

  Destination Server A manual entry box to set the target DHCP server

- Click **Save**

The DHCPv6 Relay function works identically to the DHCP Relay function for IPv4.

21.5 DNS Resolver

The DNS Resolver in pfSense® utilizes `unbound`, which is a validating, recursive, caching DNS resolver that supports DNSSEC and a wide variety of options. The DNS Resolver is enabled by default in current versions of pfSense.

By default, the DNS Resolver queries the root DNS servers directly and does not use DNS servers configured under **System > General Setup** or those obtained automatically from a dynamic WAN. This behavior may be changed, however, using the **DNS Query Forwarding** option. By contacting the roots directly by default, it eliminates many issues typically encountered by users with incorrect local DNS configurations, and the DNS results are more trustworthy and verifiable with Domain Name System Security Extensions (DNSSEC).

21.5.1 DNS Resolver Advanced Options

pfSense® software provides a GUI to configure some of the more common advanced options available in `unbound`. The options below are documented as found in the unbound.conf man page.

Hide Identity When set, attempts to query the server identity (`id.server` and `hostname.bind`) are refused.

Hide Version When set, attempts to query the server version (`version.server` and `version.bind`) are refused.

Prefetch Support When enabled, message cache elements are prefetched before they expire to help keep the cache up to date. This option can cause an increase of around 10% more DNS traffic and load on the server, but frequently requested items will not expire from the cache.

Prefetch DNS Key Support When enabled, DNSKEYs are fetched earlier in the validation process when a Delegation Signer record is encountered. This helps lower the latency of requests but utilizes a little more CPU, and requires the cache to be set above zero.

Harden DNSSEC Data If this option is disabled and no DNSSEC data is received, then the zone is made insecure. DNSSEC data is required for trust-anchored zones. If such data is absent, the zone becomes bogus.

Message Cache Size The message cache stores DNS response codes and validation statuses. The resource record set (RRSet) cache will automatically be set to twice this amount. The RRSet cache contains the actual resource record data. The default is *4 MB*.

Outgoing TCP Buffers The number of outgoing TCP buffers to allocate per thread. The default value is *10*. If set to *0*, TCP queries will not be sent to authoritative servers.

Incoming TCP Buffers The number of incoming TCP buffers to allocate per thread. The default value is *10*. If set to *0*, TCP queries will not be accepted from clients.

EDNS Buffer Size Number of bytes size to advertise as the EDNS reassembly buffer size. This value is placed in UDP datagrams sent to peers. RFC recommendation is *4096* (the default). If fragmentation reassembly problems occur, usually observed as timeouts, then a value of *1480* may help. The *512* value bypasses most MTU path problems, but it is excessive and can generate an excessive amount of TCP fallback.

Number of Queries per Thread The number of queries that every thread will service simultaneously. If additional queries arrive that need to be serviced, and no queries can be jostled out, the new queries are dropped

Jostle Timeout Timeout used when the server is very busy. This protects against denial of service by slow queries or high query rates. The default value is *200* milliseconds. Set to a value that approximates the round-trip time to the authority servers. As new queries arrive, 50% are allowed to run and 50% are replaced by new queries if they are older than the stated timeout.

Maximum TTL for RRsets and Messages The Maximum Time to Live (TTL) for RRsets and messages in the cache, specified in seconds. The default is `86400` seconds (1 day). When the internal TTL expires the cache item is expired. This can be configured to force the resolver to query for data more often and not trust (very large) TTL values

Minimum TTL for RRsets and Messages The Minimum Time to Live for RRsets and messages in the cache, specified in seconds. The default is `0` seconds. If a record has a TTL lower than the configured minimum value, data can be cached for longer than the domain owner intended, and thus less queries are made to look up the data. The `0` value ensures the data in the cache is not kept longer than the domain owner intended. High values can lead to trouble as the data in the cache may not match up with the actual data if it changes.

TTL for Host Cache Entries Time to Live, in seconds, for entries in the infrastructure host cache. The infrastructure host cache contains round trip timing, lameness, and EDNS support information for DNS servers. The default value is *15 minutes*.

Number of Hosts to Cache Number of infrastructure hosts for which information is cached. The default is *10,000*.

Unwanted Reply Threshold If enabled, a total number of unwanted replies is tracked in every thread. When the threshold is reached, a defensive action is taken and a warning is printed to the log file. The defensive action is to clear the RRSet and message caches, hopefully flushing away any poison. The default is disabled, but if enabled a value of 10 million is suggested.

Log Level Select the log verbosity. Default is *Level 1*.

Level 0 No verbosity, only errors.

Level 1 Operational information.

Level 2 Detailed operational information.

Level 3 Query level information, output per query.

Level 4 Algorithm level information.

Level 5 Logs client identification for cache misses.

Disable Auto-added Access Control Disables the automatically-added access control entries. By default, IPv4 and IPv6 networks residing on internal interfaces of this firewall are permitted. Allowed networks must be manually configured on the **Access Lists** tab if when checked.

Experimental Bit 0x20 Support Use 0x20-encoded random bits in the DNS query to foil spoofing attempts. See the implementation draft dns-0x20 for more information:

21.5.2 DNS Resolver Access Lists

Unbound requires access lists (ACLs) to control which clients are allowed to submit queries. By default, IPv4 and IPv6 networks residing on internal interfaces of this firewall are permitted. Additional networks must be allowed manually.

Note: The automatic ACLs may be disabled using the **Disable Auto-added Access Control** option on the **Advanced Settings** tab.

To manage Access Lists for the DNS Resolver, navigate to **Services > DNS Resolver**, **Access Lists** tab. From this list, new entries may be added and existing entries may be edited or deleted.

When adding or editing an entry, the following options are available:

Access List Name The name for the Access List, which appears as a comment in the access list configuration file.

Action Method for handling the networks contained in this Access List

Deny Stops queries from clients in the configured networks

Refuse Stops queries from clients in the configured networks and sends back a `REFUSED` response code

Allow Allows queries from clients in the configured networks

Allow Snoop Allows recursive and nonrecursive queries from clients in the configured networks, used for cache snooping, and typically only configured on administrative hosts.

Description A longer text field for reference notes about this entry.

Networks A list of networks to be governed by this access list entry.

21.5.3 DNS Resolver and IPv6

The DNS Resolver is fully compatible with IPv6. It accepts and makes queries on IPv6, supports AAAA records, and has no known issues with any aspect of IPv6 and handling DNS.

21.5.4 DNS Resolver Configuration

To configure the DNS Resolver, navigate to **Services > DNS Resolver**

Enable Checking this box turns on the DNS Resolver, or uncheck to disable this functionality. The DNS Forwarder and DNS Resolver cannot both be active at the same time on the same port, so disable the DNS Forwarder or move one service or the other to a different port before attempting to enable the DNS Resolver.

Listen Port By default, the DNS Resolver listens on TCP and UDP port `53`. This is normal for any DNS server, as it is the port clients will try to use. There are some cases where moving the DNS Resolver to another Listen Port, such as 5353 or 54 is desirable, and then specific sources may be forwarded there via port forwards.

Interfaces By default, the DNS Resolver listens on every available interface and IPv4 and IPv6 address. The Interface control limits the interfaces where the DNS forwarder will accept and answer queries. This can be used to increase security in addition to firewall rules. If a specific interface is selected, both the IPv4 and IPv6 addresses on that interface will be used for answering queries. The `unbound` daemon will only bind to the selected interface. Queries sent to other IP addresses on the firewall will be silently discarded.

Outgoing Network Interfaces By default the DNS Resolver utilizes all interfaces for outbound queries, so it will source the query from whichever interface and IP address is closest to the target server from a routing perspective. Selecting specific interfaces will limit the choices to only specific interfaces that may be used as a source of queries.

System Domain Local Zone Type This option determines the type of `local-zone` configured in `unbound` for the system domain. The zone type governs the type of response to give clients when there is no match in local data such as Host Overrides, DHCP hosts, etc. In each case, if there is a local match, the query is answered normally. The available types to govern non- matching responses are:

Deny Drops the query and does not answer the client.

Refuse Notifies the client that the query was refused (Using `rcode REFUSED`).

Static Returns a `NODATA` or `NXDOMAIN` response to the client.

Transparent This is the default behavior. If the query is for a name that does not exist locally, it is resolved as usual. If the name has a local match but the type is different, a `NOERROR`, `NODATA` response is sent to the client

Type Transparent Similar to transparent, it also passes through queries where the name matches but the type does not. For example, if a client queries for an AAAA record but only an A record exists, the AAAA query is passed on rather than receiving a negative response.

Redirect Handles queries from local data and redirects queries for zones underneath the local zone (e.g. subdomains). This can be used to control queries for all subdomains under the given domain.

Inform Answers normally, but logs the client query.

Inform Deny Denies and logs the query.

No default Disables any default content for the zone without affecting query behavior.

DNSSEC Enables Domain Name System Security Extensions (DNSSEC), which allows clients to trust the origin and content of DNS responses. This is enabled by default. DNSSEC protects against manipulation of DNS responses, such as DNS cache poisoning or other query interception, but it does not make the contents of responses secret. DNSSEC works best when using the root servers directly, unless the forwarding servers support DNSSEC. If upstream DNS servers do not support DNSSEC in forwarding mode or with domain overrides, DNS queries are known to be intercepted upstream, or clients have issues with over-size DNS responses, DNSSEC may need to be disabled.

DNS Query Forwarding Disabled by default. When enabled, `unbound` will use the system DNS servers from **System > General Setup** or those received from a dynamic WAN, rather than using the root servers directly. This is better for a multi-WAN scenario where fine control of DNS query routing is desired, but typically also requires disabling DNSSEC due to a lack of support by upstream DNS servers or other problems forwarding the queries.

DHCP Registration When active, internal machine names for DHCP clients can be resolved using DNS. This only works for clients that specify a hostname in their DHCP requests. The domain name from **System > General Setup** is used as the domain name on the hosts.

Static DHCP This works the same as **Register DHCP leases in DNS forwarder**, except that it registers the DHCP static mapping addresses instead.

Custom Options A text area for placing advanced directives for `unbound` that are not supported by the GUI directly. If `unbound` does not start correctly after entering custom options, add `server:` on a line before the custom options.

Host Overrides

Custom DNS entries can be created in the **Host Overrides** section of the page. Host overrides can define new records, or override existing records so that local clients receive the configured responses instead of responses from upstream DNS servers. This is also useful for split DNS configurations (see *Split DNS*), and as a semi-effective means of blocking access to certain specific websites.

Multiple records may be defined for the same hostname, and all IP addresses will be returned in the result. This can be used to supply both an IPv4 (A) and IPv6 (AAAA) result for a single hostname.

Note: We do not recommend using only the DNS override functionality as a means of blocking access to certain sites. There are countless ways to get around this. It will stop non-technical users, but it is easy to circumvent for those with more technical aptitude.

Host This field defines only the hostname portion of the DNS record (without the domain), e.g. `www`. It may be left blank to make an override record for the domain itself (Similar to an "@" record in bind.)

Domain This field is required, and defines the domain name for the override entry, e.g. `example.com`.

IP Address The IP address (either IPv4 or IPv6) to return as the result for a DNS lookup of this entry.

Description A text description used to identify or give more information about this entry.

Additional Names for This Host Defines additional hostnames for the same IP address (much like CNAME records) to keep them in a single override entry.

Domain Overrides

Domain overrides are found at the bottom of the DNS Resolver page. These entries specify an alternate DNS server to use for resolving a specific domain.

One example of where this is commonly deployed is in small business networks with a single internal server with Active Directory, usually Microsoft Small Business Server. The DNS requests for the Active Directory domain name must be resolved by the internal Windows Server for Active Directory to function properly. Adding an override for the Active Directory domain pointing to the internal Windows server IP address ensures these records are resolved properly whether clients are using this firewall as a DNS server or the Windows Server directly.

In an Active Directory environment the best practice is to have clients always use the Windows DNS server as the primary DNS server so dynamic name registration and other domain-related DNS tasks function properly. In environments with only one Windows DNS server, enable the DNS Resolver with an override for the Active Directory domain and use this firewall as the secondary DNS server for the internal machines. This ensures DNS resolution (except for Active Directory) does not have a single point of failure, and loss of the single server won't mean a complete Internet outage. The loss of a single server in such an environment will usually have significant consequences, but users will be more apt to leave the administrator alone to fix the problem if they can still check out their lolcats, Facebook, Twitter, et al in the meantime.

Another common use of DNS overrides is to resolve internal DNS domains at remote sites using a DNS server at the main site accessible over VPN. In such environments all DNS queries are typically resolved at the central site for centralized control over DNS, however some organizations prefer letting Internet DNS resolve with pfSense at each site, and only forwarding queries for internal domains to the central DNS server. Note a static route is necessary for this to function over IPsec. See *Accessing Firewall Services over IPsec VPNs* for more information.

Domain The Domain field sets the domain name that will be resolved using this entry. This does not have to be a valid TLD, it can be anything (e.g. `local`, `test`, `lab`), or it can be an actual domain name (`example.com`).

IP Address Specifies the IP Address of the DNS server to which the queries for hostnames in Domain are sent. If the target DNS server is running on a port other than `53`, add the port number after the IP address with an `@` separating the values, for example:

```
192.0.2.3@5353
```

Description A text description used to identify or give more information about this entry.

21.5.5 DNS Resolver and Multi-WAN

With the default settings, the DNS Resolver will have issues in a Multi-WAN environment. The main issue is that the DNS Resolver wants to query the root DNS servers directly. These queries will only be sent out using the default gateway. If the WAN containing the default gateway fails then DNS queries will also likely fail. There are ways to work around this limitation, however:

Forwarding Mode

Enable **DNS Query Forwarding** and configure at least one DNS server per WAN gateway under **System > General Setup**. DNSSEC may also need to be disabled, depending on upstream DNS server support.

Default Gateway Switching

Enable **Default Gateway Switching** under **System > Advanced**, **Miscellaneous** tab. This will move the default gateway to the next available gateway if the preferred default fails. However, this option is still considered experimental and may have problems in certain cases.

21.5.6 DNS Resolver and DNS Rebinding Protection

By default, DNS Rebinding protection is enabled and private IP address responses are rejected. To allow private IP address responses from a known domain, use the **Custom Options** box in the DNS Resolver settings to configure allowed domains as follows:

```
server:
private-domain: "example.com"
```

21.5.7 CLI Commands

Unbound provides various command line utilities to manage the DNS Cache server. The following control commands are currently not available in the webGUI but can be executed from the command line.

Note: Unbound does not use the default conf file location; Use the `-c` flag to tell Unbound the configuration file location:

```
unbound-control -c /var/unbound/unbound.conf <unbound-command-to-run>
```

To remove items from the cache:

`unbound-control flush <name>` Removes `<name>` from the cache, all record types which include A, AAAA, NS< SOA, CNAME, DNAME, MX, PTR, SRV and NAPTR records.

`unbound-control flush_type <name> <type>` Removes the `<name>` and `<type>` from the cache where `<type>` is a particular record type.

`unbound-control flush_zone <name>` Removes all information at or below the `<name>` from cache. For example, `.com` will remove all entries below `.com`. Note this process is slow as the entire cache must be inspected.

To determine the name servers Unbound will Query to lookup a zone:

```
unbound-control lookup <name>
```

21.6 DNS Forwarder

The DNS Forwarder in pfSense® software is a caching DNS resolver that employs the `dnsmasq` daemon. It is disabled by default in current versions, with the *DNS Resolver* (`unbound`) being active by default instead. The DNS Forwarder will remain enabled on older systems or upgraded systems where it was active previously.

The DNS Forwarder uses DNS servers configured at **System > General Setup**, or those obtained automatically from an ISP for dynamically configured WAN interfaces (DHCP, PPPoE, PPTP). For static IP address WAN connections, DNS servers must be entered at **System > General Setup** or during the setup wizard for the DNS forwarder to function. Statically configured DNS servers may also be used with dynamically configured WAN interfaces by unchecking the **Allow DNS server list to be overridden by DHCP/PPP** on WAN box on the **System > General Setup** page.

By default, the DNS Forwarder queries all DNS servers at once, and the only the first response received is used and cached. This results in much faster DNS service from a client perspective, and can help smooth over problems that stem from DNS servers which are intermittently slow or have high latency, especially in Multi-WAN environments. This behavior can be disabled by activating the **Query DNS servers sequentially** option.

21.6.1 DNS Forwarder and IPv6

The DNS Forwarder is fully compatible with IPv6. It accepts and makes queries on IPv6, supports AAAA records, and has no known issues with any aspect of IPv6 and handling DNS.

21.6.2 DNS Forwarder Configuration

To configure the DNS Forwarder, navigate to **Services > DNS Forwarder**

The available options for the DNS Forwarder are:

Enable Checking this box turns on the DNS Forwarder, or uncheck to disable this functionality. The DNS Forwarder and DNS Resolver cannot both be active at the same time on the same port, so disable the DNS Resolver or move one service or the other to a different port before attempting to enable the DNS Forwarder.

DHCP Registration When active, internal machine names for DHCP clients can be resolved using DNS. This only works for clients that specify a hostname in their DHCP requests. The domain name from **System > General Setup** is used as the domain name on the hosts.

Static DHCP This works the same as **Register DHCP leases in DNS forwarder**, except that it registers the DHCP static mapping addresses instead.

Prefer DHCP When one IP address has multiple hostnames, doing a reverse lookup may give an unexpected result if one of the hostname is in host overrides and the system uses another hostname over DHCP. Checking this option will place the DHCP obtained hostnames above the static mappings in the hosts file on the firewall, causing them to be consulted first. This only affects reverse lookups (PTR), since they only return the first result and not multiple. For example, this would yield a result of `labserver01.example.com`, a test server's DHCP obtained IP address, rather than a host override name of `testwww.example.com` that would be returned otherwise.

Query DNS servers sequentially By default, the firewall queries all DNS servers simultaneously and uses the fastest result. This isn't always desirable, especially if there is a local DNS server with custom hostnames that could by bypassed by using a faster but public DNS server. Checking this option causes queries to be made to each DNS server in sequence from the top down, and the firewall waits for a timeout before moving on to the next DNS server in the list.

Require domain Requires a domain name on hostnames to be forwarded to upstream DNS servers. Hosts without a name will still be checked against host overrides and DHCP results, but they will not be queried against the name servers configured on the firewall. Instead, if a short hostname does not exist locally, an NXDOMAIN result ("Not Found") is returned to the client.

Do not forward private reverse lookups When checked, this option prevents `dnsmasq` from making reverse DNS (PTR Record) lookups for RFC1918 private IP addresses to upstream name servers. It will still return results from local entries. It is possible to use a domain override entry for the reverse lookup zone, e.g. `1.168.192 .in-addr.arpa`, so that queries for a specific subnet will still be sent to a specific DNS server.

Listen Port By default, the DNS Forwarder listens on TCP and UDP port `53`. This is normal for any DNS server, as it is the port clients will try to use. There are some cases where moving the DNS Forwarder to another Listen Port, such as 5353 or 54 is desirable, and then specific queries may be forwarded there via port forwards.

Interfaces By default, the DNS Forwarder listens on every available interface and all available IPv4 and IPv6 addresses. The Interface control limits the interfaces where the DNS forwarder will accept and answer queries. This can be used to increase security in addition to firewall rules. If a specific interface is selected, both the IPv4 and IPv6 addresses on that interface will be used for answering queries. Queries sent to other IP addresses on the firewall will be silently discarded.

Strict Interface Binding When set, the DNS forwarder will only bind to the interfaces containing the IP addresses selected in the **Interface** control, rather than binding to all interfaces and discarding queries to other addresses. This can be used similarly to the **Listen Port** for controlling the way that the service binds so that it can coexist with other DNS services that have similar options.

Note: This option is not compatible with IPv6 in the current version of the DNS Forwarder daemon, `dnsmasq`. If this is checked, the dnsmasq process will not bind to any IPv6 addresses.

Advanced Options

Custom dnsmasq configuration parameters that are not configurable in the GUI can be placed in **Advanced Options**. For example, to set a lower TTL for DNS records, enter `max-ttl=30`. Or craft a wild card DNS record to resolve `.lab.example.com` to `192.2.5.6` by specifying `address=/lab.example.com/192.2.5.6`.

Separate commands by either a space or a newline. For more information on the possible parameters that may be used, consult the dnsmasq documentation.

Host Overrides

Host override entries provide a means to configure customized DNS entries. The configuration is identical to *Host Overrides* in the DNS Resolver, refer there for details.

Domain Overrides

Domain overrides configure an alternate DNS server to use for resolving a specific domain. The configuration is identical to *Domain Overrides* in the DNS Resolver, with some slight differences:

Domain The Domain field sets the domain name that will be resolved using this entry. This does not have to be a valid TLD, it can be anything (e.g. `local`, `test`, `lab`), or it can be an actual domain name (`example.com`).

IP Address This field can be used in one of three ways. First, it can be used to specify the IP Address of the DNS server to which the queries for hostnames in Domain are sent. Second, it can be used to override another entry by entering `#`. For example, to forward `example.com` to `192.2.66.2`, but have `lab.example.com` forward on to the standard name servers, enter a `#` in this field. Third, it can be used to prevent non-local lookups by entering a `!`. If host override entries exist for `www.example.org` and `mail.example.org`, but other lookups for hosts under *example.org* must not be forwarded on to remote DNS servers, enter a `!` in this field.

Source IP This field is optional, and primarily used to contact a DNS server across a VPN. Typically only specific local IP addresses are able to traverse a VPN, this field specifies which IP address on the firewall is used to source the DNS so the queries will pass properly.

Description A text description used to identify or give more information about this entry.

21.6.3 DNS Forwarder and Multi-WAN

The DNS Forwarder is fully compatible with Multi-WAN. Configure at least one DNS server per WAN gateway under **System > General Setup**.

21.6.4 DNS Forwarder and DNS Rebinding Protection

By default, DNS Rebinding protection is enabled and private IP address responses are rejected. To allow private IP address responses from a known domain, use the **Advanced Options** box in the DNS Forwarder settings to configure allowed domains as follows:

```
rebind-domain-ok=/example.com/
```

21.7 Dynamic DNS

The Dynamic DNS client built into pfSense® software registers the IP address of a WAN interface with a variety of dynamic DNS service providers. This is used to remotely access services on hosts that have WANs with dynamic IP addresses, most commonly VPNs, web servers, and so on.

Any number of Dynamic DNS clients may be configured using any of over 20 different Dynamic DNS providers, or even custom Dynamic DNS providers. Dynamic DNS clients can use any WAN, and can even register the real public IP address in environments where the firewall receives a private IP address for its WAN and is NATed upstream.

In addition to the typical HTTP/HTTPS-based Dynamic DNS providers, pfSense also supports RFC 2136 style Dynamic DNS updates directly to DNS servers.

21.7.1 Configuring a Dynamic DNS Client

pfSense® software allows registration with many different dynamic DNS providers. The available providers may be viewed by clicking the **Service Type** selector. More information about the providers may be found by searching for their name to find their web site. Several offer a basic level service at no cost, and some offer additional premium services at a cost. There is also a *Custom* option that allows for a custom URL to accommodate an unsupported provider.

Select a provider, visit their website, register for an account, and setup a hostname. The procedures for this vary with each provider, but they all have instructions on their websites. After configuring a hostname with a provider, configure pfSense with matching settings.

Most providers have the same, or similar options. There are a few types with custom options that will be covered later in this section.

To configure a Dynamic DNS client:

- Navigate to **Services > Dynamic DNS**
- Click  **Add** to add a new entry
- Configure the options as follows:

  Disable Check to disable the entry, or leave unchecked so it will be active.

  Service Type Select the dynamic DNS provider here.

Interface to Monitor Select the interface that has the IP address to keep updated, such as WAN, or an OPTx interface. Selecting a gateway group for the interface allows the Dynamic DNS entry to switch between WANs so it can allow inbound Multi-WAN failover of services on this hostname.

Hostname Enter the hostname created at the dynamic DNS provider. This is typically the complete fully qualified domain name, such as `myhost.example.com`, except for Namecheap where this is only the host portion of the address.

Domain Name For Namecheap hosts, this box must be set to the domain part of the full hostname.

MX An MX (Mail Exchanger) record is how Internet mail servers know where to deliver mail for a domain. Some dynamic DNS providers will let MX records be configured via the dynamic DNS client. If the chosen provider allows this, enter the host name of the mail server that will receive Internet mail for the dynamic DNS domain.

Wildcards When wildcard DNS is enabled on a dynamic DNS name, all host name queries under the given domain will resolve to the IP address of the dynamic DNS host name. For example, if the host name is `example.dyndns.org`, enabling wildcard will make `*.example.dyndns.org` (`a.example.dyndns.org`, `b.example.dyndns.org`, etc.) resolve the same as `example.dyndns.org`.

Verbose Logging Check this option to increase the logging for the Dynamic DNS update process, which is useful for troubleshooting update problems.

Verify SSL Peer When checked, the SSL certificate of the DynDNS provider server will be validated. Some servers with self-signed certificates, or those using a less common CA, may require this to be set.

Username Enter the username for the dynamic DNS provider. Provider-specific requirements:

Namecheap, FreeDNS Leave blank

Route 53 Enter the **Access Key ID**

GleSYS Enter the **API user**

Custom The username is used with basic HTTP authentication and may be left blank.

Password Enter the password for the dynamic DNS provider. Provider-specific requirements:

Namecheap, FreeDNS This is the **Authentication Token**

Route 53 Enter the **Secret Access Key**

GleSYS Enter the **API Key**

DNSimple Enter the **API Token**

Description A text field for reference.

- Click **Save**

Providers with Extra or Different Settings

Some providers have special settings or certain fields that need to be set in a specific way that may not be obvious. The differences are outlined in this section.

Namecheap

As mentioned earlier in the settings above, Namecheap requires that the fully qualified domain name be split into the hostname part and domain name part in separate fields.

When setting up Dynamic DNS for a *Namecheap* domain, an authentication token is given by Namecheap. This goes in the **Password** field, and the **Username** field is left blank.

HE.net Tunnelbroker

The *HE.net Tunnelbroker* choice updates an IPv6 tunnel endpoint IP address when the WAN IP changes. The **Hostname** in this case is the **Tunnel ID** from HE.net.

Route 53

When using an Amazon *Route 53* type, the Username is the **Access Key ID** provided by Amazon.

The following additional options are available when using *Route 53*:

Verify SSL Peer Enable to verify the server certificate when using HTTPS

Zone ID Received when creating the domain in Route 53. Must be filled in.

TTL Time to Live for the DNS record.

Custom

The *Custom* Dynamic DNS type configures options that allow for updating otherwise unsupported services. When using the custom Dynamic DNS type, the **Username** and **Password** fields are sent using HTTP basic authentication.

The following additional options are available when using *Custom*:

Interface to send update from Almost always the same as the Interface, but can be changed as needed.

Force IPv4 Resolving When checked, the update host will only be resolved using IPv4

Verify SSL Peer Enable to verify the server certificate when using HTTPS

Update URL The URL given by the Dynamic DNS provider for updates. If the IP address must appear in the URL, enter it as `%IP%` and the real value will be substituted as needed.

Result Match Defines expected output from the Dynamic DNS query. If it succeeds and matches the output given, then pfSense will know that the update was successful. If it does not match exactly, then it is assumed that the update failed. Leave empty to disable result checking.

DNSSimple

Verify SSL Peer Enable to verify the server certificate when using HTTPS

Zone ID Received when creating the domain.

TTL Time to Live for the DNS record.

21.7.2 Configuring RFC 2136 Dynamic DNS updates

RFC 2136 Dynamic DNS registers a hostname on any DNS server supporting RFC 2136 style updates. This can be used to update DNS records on BIND and Windows Server DNS servers, amongst others.

RFC 2136 Dynamic DNS entries may be used at the same time as regular style Dynamic DNS service providers, and like those, any number of entries can be created. RFC 2136 will update the A record, and the AAAA record if IPv6 is configured on the monitored interface.

See also:

Configuring the server infrastructure for RFC 2136 Dynamic DNS hosting is beyond the scope of this documentation, but there is a basic how-to in the recipes section: *Configuring BIND as an RFC 2136 Dynamic DNS Server*.

To configure an RFC 2136 Dynamic DNS client:

- Navigate to **Services > Dynamic DNS**
- Click the **RFC 2136** tab
- Click  **Add** to add a new entry
- Configure the options as follows:

  Enable Controls whether or not the entry is active. If it is unchecked, updates will not be performed for this entry.

  Interface The IP address on the chosen interface will be sent when performing the DNS update.

  Hostname The fully qualified domain name (FQDN) of the dynamic DNS entry to update. For example, `myhost.example.com`.

  TTL The Time To Live for the DNS entry, in seconds. Higher values will be cached longer by other name servers, so lower values are better to be sure that DNS updates are picked up in a timely manner by other servers. Usually a value between `30` and `180` seconds is reasonable, depending on how often the IP address changes.

  Key Name The name of the key as specified in the DNS server configuration. For Host keys, this is typically the FQDN, so it would be identical to the value in the **Hostname** field. For Zone keys this would be the name of the DNS zone.

  Key Type Can be one of *Zone*, *Host* or *User*. The type of key is determined by the server, so consult the server configuration or the DNS server administrator to determine the **Key Type**. Typically this is set to *Host*.

  Key Contains the actual text of the key, e.g. `/0/4bxF9A08n/zke/vANyQ==`. This value is generated by the DNS server or administrator.

  Server The IP address or hostname of the DNS server to which updates are sent.

  Protocol When unchecked, the DNS update is sent over UDP, when checked it uses TCP instead.

Use Public IP By default, the interface IP address is always sent to the name server for the DNS update. If this box is checked, when a private IP address is detected on the selected **Interface**, a check is done to determine what the actual public IP address is, and then that IP address is used for the DNS update.

Record Type Determines which record(s) will be updated for this entry. For the IPv4 address, use *A*, for IPv6, use *AAAA*, or choose *Both*.

Description A free-text description of the entry for reference.

As with the other Dynamic DNS types, RFC 2136 updates are performed only when an IP address change is detected, or once every 25 days.

21.7.3 Configuring IP Address Check Services for Dynamic DNS

pfSense® software version 2.3.3 and later support custom IP address check services. These services are used by Dynamic DNS clients to determine the public IP address of the firewall when a WAN interface is behind an upstream NAT device.

To create or edit one of these services, navigate to **Services > Dynamic DNS** on the **Check IP Services** tab.

Settings

Fill out the form fields on the page as follows:

- **Enable**: Allow this service to be used by Dynamic DNS clients
- **Name**: A short name to identify this service
- **URL**: The full URL to the IP address check page
- **Username/Password**: Optional authentication to use when accessing the URL
- **Verify SSL Peer**: Check this box if the server has a self-signed SSL certificate or a certificate from a CA that is not trusted by the firewall.
- **Description**: A longer description of this service

Once a service is defined, it may be selected on individual Dynamic DNS service entries.

Server-Side Configuration Examples

Hosting one of these services is very simple. The server page need only print the requesting client IP address in the expected format:

```
Current IP Address: x.x.x.x
```

nginx (internal/native)

```
location /ip {
    default_type text/html;
    return 200 "<html><head><title>Current IP Check</title></head><body>Current IP␣
↪Address: $remote_addr</body></html>";
}
```

nginx (internal with LUA)

```
location = /ip {
    default_type text/html;
    content_by_lua '
        ngx.say("<html><head><title>Current IP Check</title></head><body>Current IP␣
↪Address: ")
        ngx.say(ngx.var.remote_addr)
        ngx.say("</body></html>")
    ';
}
```

PHP

```
<html><head><title>Current IP Check</title></head><body>Current IP Address: <?=$_
↪SERVER['REMOTE_ADDR']?></body></html>
```

21.8 SNMP

The Simple Network Management Protocol (SNMP) daemon enables remote monitoring of some pfSense® software parameters. Depending on the options chosen, monitoring may be performed for network traffic, network flows, pf queues, and general system information such as CPU, memory, and disk usage. The SNMP implementation is bsnmpd, which by default only has the most basic management information bases (MIBs) available, and is extended by loadable modules. In addition to acting as an SNMP daemon, it can also send traps to an SNMP server for certain events. These vary based on the modules loaded. For example, network link state changes will generate a trap if the MIB II module is loaded.

The SNMP service can be configured by navigating to **Services > SNMP**.

The easiest way to see the available data is to run `snmpwalk` against the firewall from another host with `net-snmp` or an equivalent package installed. The full contents of the MIBs available are beyond the scope of this documentation, but there are plenty of print and online resources for SNMP, and some of the MIB trees are covered in RFCs. For example, the Host Resources MIB is defined by RFC 2790.

See also:

The Hangouts Archive contains a video which covers monitoring via SNMP.

21.8.1 SNMP and IPv6

The `bsnmpd` daemon does not currently support IPv6.

21.8.2 SNMP Daemon

These options dictate if, and how, the SNMP daemon will run. To turn the SNMP daemon on, check **Enable**. Once **Enable** has been checked, the other options may then be changed.

Polling Port SNMP connections are made using only UDP, and SNMP clients default to using UDP port 161. This setting controls which port is used for the SNMP daemon, and the SNMP client or polling agent must be changed to match.

System location This text field specifies a string to return when the system location is queried via SNMP. Any text may be used here. For some devices a city or state may be close enough, while others may need more specific detail such as which rack and position in which the system resides.

System contact A string defining contact information for the system. It can be a name, an e-mail address, a phone number, or whatever is needed.

Read Community String With SNMP, the community string acts as a kind of username and password in one. SNMP clients will need to use this community string when polling. The default value of `public` is common, so we strongly recommend using a different value in addition to restricting access to the SNMP service with firewall rules.

21.8.3 SNMP Traps

To instruct the SNMP daemon to send SNMP traps, check **Enable**. Once Enable has been checked, the other options may then be changed.

Trap server The trap server is the hostname or IP address to which SNMP traps are forwarded.

Trap server port By default, SNMP traps are set on UDP port `162`. If the SNMP trap receiver is set for a different port, adjust this setting to match.

SNMP trap string This string will be sent along with any SNMP trap that is generated.

21.8.4 Modules

Loadable modules allow the SNMP daemon to understand and respond to queries for more system information. Each loaded module will consume additional resources. As such, ensure that only required modules are loaded.

MibII This module provides information specified in the standard MIB II tree, which covers networking information and interfaces. Having this module loaded will, among other things, provides network interface information including status, hardware and IP addresses, the amount of data transmitted and received, and much more.

Netgraph The netgraph module provides some netgraph-related information such as netgraph node names and statuses, hook peers, and errors.

PF The pf module provides a wealth of information about pf. The MIB tree covers aspects of the ruleset, states, interfaces, tables, and ALTQ queues.

Host Resources This module provides information about the host itself, including uptime, load average and processes, storage types and usage, attached system devices, and even installed software. This module requires MibII, so if MibII is unchecked when this option is checked, MibII will be checked automatically.

UCD This module provides various system information knows as the ucdavis MIB, or UCD-SNMP-MIB. It provides information about memory usage, disk usage, running programs, and more.

Regex The Regex module is reserved for future use or use by users customizing the code to their needs. It allows creating SNMP counters from log files or other text files.

21.8.5 Interface Binding

This option configures the SNMP daemon to listen only on the chosen interface or virtual IP address. All interfaces with IP addresses, CARP VIPs, and IP Alias VIPs are displayed in the drop-down list.

Binding to a specific local interface can ease communication over VPN tunnels, as it eliminates the need for the previously mentioned static route, and it also provides extra security by not exposing the service to other interfaces. It can also improve communication over multiple local interfaces, since the SNMP daemon will reply from the "closest" address to a source IP address and not the IP address to which the query was sent.

21.9 UPnP & NAT-PMP

Universal Plug and Play (UPnP) and NAT Port Mapping Protocol (NAT-PMP) are network services which allow software and devices to configure each other when attaching to a network. This includes automatically creating their own dynamic NAT port forwards and associated firewall rules.

The UPnP and NAT-PMP service on pfSense®, found at **Services > UPnP & NAT-PMP**, enables client PCs and other devices such as game consoles to automatically allow required inbound traffic. There are many popular programs and systems which support UPnP, such as Skype, uTorrent, mIRC, IM clients, Wii U, PlayStation 4, and XBox One. NAT-PMP is supported on Apple products.

UPnP employs the Simple Service Discovery Protocol (SSDP) for network discovery, which uses UDP port 1900. The UPnP daemon used by pfSense, miniupnpd , also uses TCP port 2189. When using a strict LAN ruleset, manually add firewall rules to allow access to these services, especially if the default LAN-to-any rule has been removed, or in bridged configurations. NAT-PMP is also handled by miniupnpd and uses UDP port 5351.

21.9.1 UPnP & NAT-PMP and IPv6

As of this writing, the UPnP and NAT-PMP service on current versions of pfSense supports IPv6, but client support is still spotty.

21.9.2 Security Concerns

UPnP and NAT-PMP are a classic example of the "Security vs. Convenience" trade-off. By their very nature, these services are insecure. Any program on the network can allow in and forward any traffic – a potential security nightmare. On the other side, it can be a chore to enter and maintain NAT port forwards and their associated rules, especially when it comes to game consoles. There is a lot of guesswork and research involved to find the proper ports and settings, but UPnP *just works* and requires little administrative effort. Manual port forwards to accommodate these scenarios tend to be overly permissive, potentially exposing services that should not be open from the Internet. The port forwards are also always on, where UPnP may be temporary.

Access controls exist in the UPnP service configuration, which helps to lock down which devices are allowed to make alterations. Over and above the built-in access controls, further control may be exerted with firewall rules. When properly controlled, UPnP can also be a little more secure by allowing programs to pick and listen on random ports, instead of always having the same port open and forwarded.

21.9.3 Configuration

To configure UPnP and NAT-PMP:

- Navigate to **Services > UPnP & NAT-PMP**
- Configure the options as follows:

  Enable UPnP & NAT-PMP Master control for the entire service. When unchecked, all of the services on this page are disabled.

  Allow UPnP Port Mapping When checked, UPnP is allowed.

  Allow NAT-PMP Port Mapping When checked, NAT-PMP is allowed.

  External Interface The WAN interface for outgoing traffic. This must be set to the WAN containing the default gateway. Only one **External Interface** may be selected.

  Interfaces The local interfaces where clients allowed to use UPnP/NAT-PMP reside. When a bridge is in use, only select the bridge interface with an IP address. Multiple interfaces may be selected.

  Download Speed Maximum download speed reported to clients, in Kilobits per second.

  Upload Speed Maximum upload speed reported to clients, in Kilobits per second.

  Override WAN Address Selects an alternate interface IP address to use, such as a CARP or IP Alias Virtual IP address.

  Traffic Shaping Queue The name of an ALTQ (not Limiter) traffic shaping queue in which traffic allowed through using UPnP will be placed.

  Note: Exercise caution when selecting this queue. UPnP is used by traffic such as game consoles, which need high priority, and also by file transfer clients which may need low priority.

  Log Packets When checked, port forwards generated by UPnP/NAT-PMP will be set to log, so that each connection made will have an entry in the firewall logs, found at **Status > System Logs**, on the **Firewall** tab.

  Use System Uptime By default, the UPnP daemon reports the service uptime when queried rather than the system uptime. Checking this option will cause it to report the actual system uptime instead.

  Deny Access by Default When checked, UPnP will only allow access to clients matching the access rules. This is a more secure method of controlling the service, but as discussed above, is also less convenient.

  User Specified Permissions These fields specify user-defined access rules. If the default-deny option is chosen, rules must be set to allow access. Additional rules may be added by clicking  **Add** Rules are formulated using the following format:

```
<[allow|deny]> <[external port|port range]> <[internal IP|IP/CIDR]> <[internal␣
↪port|port range]>
```

- Click **Save**

The UPnP and/or NAT-PMP service will be started automatically.

UPnP User Permission Examples

Deny access to external port `80` forwarding from everything on the LAN, `192.168.1.1`, with a `/24` subnet, to local port `80`:

```
deny 80 192.168.1.1/24 80
```

Allow `192.168.1.10` to forward any unprivileged port:

```
allow 1024-65535 192.168.1.10 1024-65535
```

21.9.4 Status

The status of the UPnP daemon process may be viewed at **Status > Services**. The Service Status page shows if the daemon is running or stopped, and allows the service to be stopped, started or restarted. Under normal circumstances, manually managing the daemon is not necessary.

A list of currently forwarded ports and clients, similar to Figure *UPnP & NAT-PMP Status Screen Showing Client PCs With Forwarded Ports*, may be viewed under **Status > UPnP & NAT-PMP**.

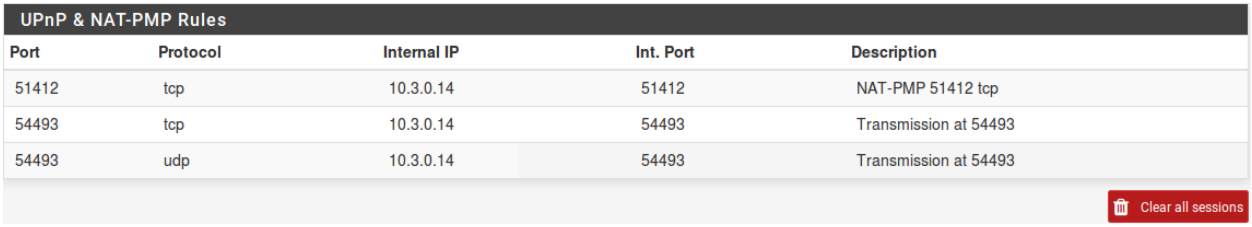


Fig. 1: UPnP & NAT-PMP Status Screen Showing Client PCs With Forwarded Ports

21.9.5 Troubleshooting

Most issues with UPnP tend to involve bridging. In this case it is important to have firewall rules allow UPnP on UDP port 1900. Since it is multicast traffic, the destination will be the broadcast address for the subnet, or in some cases making it *any* will be necessary. Consult the firewall logs at **Status > System Logs**, on the **Firewall** tab to see if traffic is being blocked. Pay particular attention to the destination address, as it may be different than expected.

Further trouble with game consoles may also be alleviated by switching to manual outbound NAT and enabling Static Port. See *Static Port* for more details.

21.10 NTPD

The NTP service is a Network Time Protocol (NTP) daemon which will listen for requests from clients and allow them to synchronize their clock with that of the pfSense® firewall. By running a local NTP server and using it for local clients, it reduces the load on the lower-stratum servers and can ensure that local systems can always reach a time server. Before delegating this task to a firewall running pfSense, the best practice is to ensure that the firewall has an accurate clock and keeps time reasonably.

21.10.1 NTP and IPv6

The NTP Project daemon fully supports IPv6 as a client and a server.

NTP Server Configuration

To configure the NTP Server:

- Navigate to **Services > NTP**
- Configure the settings as follows:

Interface Select the interface(s) to use for NTP. The NTP daemon binds to all interfaces by default to receive replies properly. This may be minimized by selecting at least one interface to bind, but that interface will also be used to source the NTP queries sent out to remote servers, not only to serve clients. Deselecting all interfaces is the equivalent of selecting all interfaces.

Time Servers A list of servers to query in order to keep the clock of this firewall synchronized. This list is initially pulled from the entries under **System > General Setup**. For best results, we recommend using at least three servers, but no more than five. Click  **Add** to configured additional time servers.

Prefer When checked, this NTP server entry is favored by the NTP daemon over others.

No Select When checked, this NTP server is not used for time synchronization, but only to display statistics.

Orphan Mode Orphan mode uses the system clock when no other clocks are available, otherwise clients will not receive a response when other servers are unreachable. The value entered here is the stratum used for **Orphan Mode**, and is typically set high enough that live servers are preferred. The default value is `12`.

NTP Graphs Check to enable RRD graphs for NTP server statistics.

Logging When logging options are active, NTP logs are written using syslog and may be found under **Status > System Logs**, on the **NTP** tab.

Log Peer Messages When checked, NTP will log messages about peer events, information, and status.

Log System Messages When checked, NTP will log messages about system events, information, and status.

Statistics Logging Click  **Show Advanced** to view these options. When enabled, NTP will create persistent daily log files in `/var/log/ntp` to keep statistics data. The format of the statistics records in the log files can be found in the ntp.conf man page

Log reference clock statistics When checked, NTP records clock driver statistics on each update.

Log clock discipline statistics When checked, NTP records loop filter statistics on each update of the local clock.

Log NTP Peer Statistics When checked, NTP records statistics for all peers of the NTP daemon, along with special signals.

Leap Seconds Click  **Show Advanced** to view these options. Defines the contents of the Leap Second file, used by NTP to announce upcoming leap seconds to clients. This is typically used only by stratum 1 servers. The exact format of the file may be found on the IETF leap second list

- Click **Save**

Access Restrictions

Access restrictions (ACLs) are configured on the **ACL** tab under **Services > NTP**. These ACLs control how NTP interacts with clients.

Default Access Restrictions Control behavior for all clients by default.

Kiss-o'-Death When set, NTP will send a KoD packet when an access violation occurs. Such packets are rate limited and no more than one per second will be sent.

Modifications When set, `ntpq` and `ntpdc` queries that attempt to change the configuration of the server are denied, but informational queries are returned.

Queries When set, all queries from `ntpq` and `ntpdc` are denied.

> **Warning:** Setting this will effectively disable the NTP status page, which relies on `ntpq`.

Service When set, NTP will deny all packets except queries from `ntpq` and `ntpdc`.

Peer Association When set, NTP denies packets that would result in a new peer association, including broadcast and symmetric active packets for peers without an existing association.

Trap Service When set, NTP will not provide mode 6 control message trap service, used for remote event logging.

Custom Access Restrictions Defines the behavior for specific client addresses or subnets. Click  **Add** to add a new network definition.

Network/mask The subnet and mask to define the client controlled by the restrictions in this entry.

Restrictions The option names are abbreviated versions of those in the default list, in the same order.

Click **Save** to store the ACLs.

Serial GPS

If this firewall has an available serial port, a Serial GPS may be used to provide a reference clock for the firewall. If the GPS also supports a Pulse Per Second (PPS) signal, that may also be used as a PPS clock reference.

Warning: USB GPS units may function, but we do not recommend their use due to USB timing issues. The overhead of USB makes its unreliable as a clock or timing source.

For best results, we recommend configuring at least two NTP servers under **System > General Setup** or **Services > NTP** to avoid loss of sync if the GPS data is not valid over time. Otherwise the NTP daemon may only use values from the unsynchronized local clock when providing time to clients.

To configure a GPS for use by NTP:

- Navigate to **Services > NTP**
- Click the **Serial GPS** tab
- Configure the settings as follows:

  GPS Type Select the make and model of the GPS unit. If the model is unknown, use the *Default* choice. If the model is known but not listed, use *Custom*.

  Serial Port All serial ports detected on the firewall are listed. Select the port with the GPS attached. On-board hardware serial ports start with `cuau`, USB serial ports are prefixed with `cuaU`.

  Baud Rate Enter the serial speed for the GPS, typically a low value such as `4800`

  NMEA Sentences By default, NTP will listen for all supported NMEA sentences. To limit this to specific types, select them from the list.

  Fudge Time 1 Specifies a constant to be added to the GPS PPS signal as an offset.

  Fudge Time 2 Specifies a constant to be added to the GPS time as an offset.

  Stratum Used to configure the stratum of the GPS clock. The default value is `0` so the GPS is preferred over all others. If another clock must be preferred instead, set the stratum value higher than the stratum of the preferred clock.

  Flags These options provide additional tweaks to fine-tune the GPS behavior:

    Prefer this clock Marks the reference clock as preferred by NTP.

    Do not use this clock Prevents the clock from being used by NTP for time synchronization, it is only displayed for reference.

    PPS signal processing Enables processing of the Pulse Per Second (PPS) signal in the GPS driver. Only enable this if the GPS is known to output a usable PPS signal.

    Falling edge PPS signal processing When set, the falling edge of the PPS signal is used for timing, rather than the rising edge.

    Kernel PPS clock discipline When set, the OS Kernel will use PPS directly for timing.

    Obscure location in timestamp Obscures the GPS data so the location of the clock cannot be determined.

    Log the sub-second fraction of the received time stamp When checked, this can rapidly fill the log, but can be useful for fine tuning of **Fudge Time 2**.

  Clock ID A 1-4 character identifier used to change the GPS Clock ID. The default value is `GPS`.

GPS Initialization Contains the initialization string sent to the GPS at start up to configure its behavior. When using the *Custom* GPS type, a proper initialization string for the GPS must be entered manually.

NMEA Checksum Calculator Calculates a checksum for use when crafting new **GPS Initialization** values or adjusting existing values.

- Click **Save**

PPS Source (Non-GPS)

A non-GPS PPS Source, such as a radio, may also be used for clock timing. It cannot be used for synchronization since there is no time data, but it can be used to ensure a clock ticks accurately.

To configure a Non-GPS PPS source:

- Navigate to **Services > NTP**
- Click the **PPS** tab
- Configure the settings as follows:

Serial Port All serial ports detected on the firewall are listed. Select the port with the GPS attached. On-board hardware serial ports start with `cuau`, USB serial ports are prefixed with `cuaU`.

Fudge Time 1 Specifies a constant to be added to the PPS signal as an offset, to account for delay between the transmitter and receiver.

Stratum Used to configure the stratum of the PPS source. The default value is `0` so the PPS source is preferred over all others. If another clock must be preferred instead, set the stratum value higher than the stratum of the preferred clock.

Flags

Falling edge PPS signal processing When set, the falling edge of the PPS signal is used for timing, rather than the rising edge.

Kernel PPS clock discipline When set, the OS Kernel will use PPS directly for timing.

Record a timestamp Record a timestamp once for each second, which is useful for constructing Allan deviation plots.

Clock ID A 1-4 character identifier used to change the PPS Clock ID. The default value is `PPS`.

- Click Save

See also:

- *Status*

21.11 Wake on LAN

The Wake on LAN (WOL) page at **Services > Wake on LAN** can wake up computers from a powered-off state by sending special "Magic Packets".

The network interface card in the client computer that is to be woken up must support WOL and it must be configured properly. Typically there is a BIOS setting to enable WOL, and non-integrated adapters often require a WOL cable connected between the NIC and a WOL header on the motherboard.

WOL has many potential uses. Typically, workstations and servers are kept running because of services they provide, files or printers they share, or for convenience. Using WOL would allow these to remain in a sleep state to conserve

power. When a service is required, the system can be woken up when needed. Another example would be if someone needs remote access to a system, but the user shut it down before leaving the office. Using WOL the target system can be awoken, and it may then be accessed once it has booted.

Warning: WOL offers no inherent security. Any system on the same layer 2 network may transmit a WOL packet, and the packet will be accepted and obeyed. It is best to only configure WOL in the BIOS for machines that need it, and disable it in all others. There are some vendor-specific WOL extensions that provide extra security, but nothing universally supported.

21.11.1 Wake Up a Single Machine

To wake up a single machine:

- Navigate to **Services > Wake on LAN**
- Select the **Interface** through which the target system can be reached
- Enter the target system **MAC address** in the format of `xx:xx:xx:xx:xx:xx`
- Click **Send**

pfSense® software will transmit a WOL Magic Packet out the chosen interface, and if everything went as planned, the system will power on and start to boot. Keep in mind that systems will take some time to boot. It may be several minutes before the target system is available.

21.11.2 Storing MAC Addresses

To store a MAC address for convenience:

- Navigate to **Services > Wake on LAN**
- Click  **Add** under the list of stored MAC addresses to add a new entry
- Select the **Interface** through which the target system can be reached
- Enter the target system **MAC address** in the format of `xx:xx:xx:xx:xx:xx`
- Enter a **Description** for the entry, such as the target system's name, owner, or location. For example: "Pat's PC" or "Sue's Server"
- Click **Save**

Once saved, the entry will be available on the list at **Services > Wake on LAN**.

Maintaining the entries is similar to other tasks in pfSense: Click  to edit an existing entry, and click  to remove an entry.

21.11.3 Wake a Single Stored Machine

To send a WOL Magic Packet to a system that has been previously stored:

- Navigate to **Services > Wake on LAN**
- Locate the desired entry in the list
- Click its **MAC address** or click the  icon in the **Actions** column

The WOL page will reload, and the Magic Packet will be sent. The status of the WOL attempt will also be displayed.

21.11.4 Wake All Stored Machines

To send a WOL Magic Packet to all stored systems at once:

- Navigate to **Services > Wake on LAN**
- Click  **Wake All Devices** under the list of stored addresses.

21.11.5 Wake from DHCP Leases View

To send a WOL Magic Packet from the DHCP Leases view:

- Navigate to **Status > DHCP Leases**
- Locate the desired system in the list
- Click  at the end of the lease row to send a WOL Magic Packet

---

Note: The WOL function is only available for systems marked **offline**, meaning they are not in the ARP table on the firewall. If a system was very recently powered off, it can take a few minutes for the ARP entry to expire before it will be marked offline.

If a system has been powered off for quite some time, clicking  **Show all configured leases** might be required to see the previous lease.

When the link is clicked, the browser will return to the WOL page, and the Magic Packet will be sent.

21.11.6 Save from DHCP Leases View

A MAC address and hostname may be copied to a new WOL mapping entry while viewing the DHCP leases.

- Navigate to **Status > DHCP Leases**
- Locate the desired system in the list
- Click  at the end of lease entry
- Confirm the values on the page, and enter any missing information.

- Click **Save**

21.12 PPPoE Server

pfSense® software can act as a PPPoE server, accepting and authenticating connections from PPPoE clients on a local interface, in the role of an access concentrator (LAC). This feature can be used to force users to authenticate before gaining network access, or otherwise control their login behavior.

The PPPoE Server is located at **Services > PPPoE Server**. The configuration is very similar to the L2TP VPN server (*L2TP VPN*).

Multiple PPPoE servers may be configured on separate interfaces. To begin setting up a PPPoE server:

- Navigate to **Services > PPPoE Server**
- Click  **Add** to add a new server entry
- Configure the PPPoE Server as follows:

  Enable When checked, this PPPoE Server instance will be active.

  Interface The single interface upon which PPPoE service will be available.

  Total User Count Determines how many clients in total are allowed to connect to this instance.

  User Max Logins Determines how many times a single client may login concurrently.

  Server Address The IP address which the pfSense system will send to the PPPoE clients to use as their gateway.

  > **Warning:** This IP address **must not** be an IP address currently in use on the firewall.

  Remote Address Range The IP address for the start of the PPPoE client subnet. Together with the Subnet Mask it defines the network used by the PPPoE clients.

  Subnet Mask Defines the CIDR mask assigned to PPPoE clients.

  Description Optional explanatory text for this server instance.

  DNS Servers Optional fields used to send specific DNS servers to the PPPoE clients, otherwise the firewall IP address will be sent to the client for DNS if the DNS Forwarder or DNS Resolver are enabled. If the DNS Forwarder and DNS Resolver are both disabled, then the DNS servers configured on the firewall will be sent instead.

- Configure **RADIUS** if that will be utilized for user authentication. Any RADIUS server may be used.

  See also:

  See *Authenticating from Active Directory using RADIUS/NPS* for information on setting up RADIUS on a Windows server.

  Use RADIUS Authentication Check to configure the PPPoE server to use at least one RADIUS server for Authentication instead of local users.

  Use RADIUS Accounting Optional, sends RADIUS accounting data to the RADIUS server to note items such as login and logout times, and bandwidth used.

  Use a Backup RADIUS Authentication Server A second RADIUS server to use if the primary RADIUS server fails.

NAS IP Address Optional, sends a specific IP address to the RADIUS server for the NAS-IP-Address attribute.

RADIUS Accounting Update The interval at which accounting data is sent to the RADIUS server, in seconds.

RADIUS Issued IP Addresses When checked, IP addresses can be assigned to users via RADIUS reply attributes.

Primary RADIUS Server The preferred RADIUS server to use for Authentication.

IP Address The IP address of the RADIUS server

Authentication Port The port used for authentication (typically `1812`)

Accounting Port The port used for accounting data (typically `1813`)

Primary RADIUS Server Shared Secret The shared secret configured for this firewall on the RADIUS server. The same value must be entered in the Confirm box.

Secondary RADIUS Server Same type of settings as the primary, but defines the secondary RADIUS server.

- Add users to the server to utilize local authentication:
  - Click  **Add User**

    Username The username for the user account

    Password The password for the user account

    IP Address An optional static IP address to assign the user at login
  - Repeat as needed
- Click **Save**

21.13 IGMP Proxy

The Internet Group Management Protocol (IGMP) Proxy provides a means to proxy multicast traffic between network segments.

The IGMP Proxy service can be found at **Services > IGMP Proxy**.

For a working IGMP Proxy configuration, one upstream and at least one downstream interface must be defined.

To configure the IGMP Proxy:

- Navigate to **Services > IGMP Proxy**
- Click  **Add** to create a new interface instance
- Configure the instance as follows:

  Interface The interface to be used for this instance

  Description Optional text to describe this instance

  Type The type of network interface defined by this instance

  Upstream Interface The outgoing interface which is responsible for communicating to available multicast data **sources**. There can only be **one** upstream interface.

Downstream Interface The distribution interfaces to the **destination** networks, where multicast **clients** can join groups and receive multicast data. One or more downstream interfaces must be configured.

Threshold The TTL threshold for forwarded data on an interface, to prevent looping from occurring. Packets with a TTL lower than the value in this field will be ignored. The default TTL is `1` if the field is left blank.

Networks A list of CIDR-masked Network entries to control what subnets are allowed to have their multicast data proxied. Click  **Add Network** to enter additional networks.

– Click **Save**

A firewall rule is also required on the **Downstream** side (e.g. *LAN*) to match and pass the multicast traffic. In the *Advanced Options* of the firewall rule, **Allow packets with IP Options** must be enabled.

The base install of pfSense® software includes services which add fundamental functionality and flexibility to the firewall. The topics in this chapter discuss services in the base installation that the firewall provides for other hosts on the network. These services include allocating IPv4 and IPv6 addresses via DHCP, DNS resolution and Dynamic DNS, SNMP, UPnP and more. Additional services can also be added with packages.

CHAPTER

TWENTYTWO

DHCP

Dynamic Host Configuration Protocol (DHCP), allows a device such as pfSense® software to dynamically allocate IP addresses to clients from a predefined pool of addresses. DHCP also sends configuration information to clients such as a gateway, DNS servers, domain name, and other useful settings.

22.1 Using DHCP Search Domains on Windows DHCP Clients

The DNS Search Domain functionality present in the DHCP Server settings in pfSense® software is only supported by some DHCP clients; pfSense software uses the standard DHCP option 119 mechanism to deliver the search domains to clients which request them.

Unfortunately, The Microsoft Windows DHCP client **does not support** requesting option 119, so no matter which DHCP server is used, clients running Microsoft Windows can never receive or use a search domain list from DHCP.

If the settings must be used by clients, they can be pushed via GPO or in the extreme case, the clients can be replaced by ones which support option 119 such as BSD, Linux, OSX, and so on.

Sources:

- http://social.technet.microsoft.com/Forums/en-US/winserverNIS/thread/9ba77f86-4708-42ca-a193-2a01b813ec27/
- http://social.technet.microsoft.com/Forums/en-US/winserverNIS/thread/7ba59619-3484-43fa-8585-a2d69ccd00df/
- http://technet.microsoft.com/en-us/library/dd572752%28v=office.13%29.aspx (See comments)
- http://serverfault.com/questions/37417/which-dhcp-client-os-support-dhcp-option-119-domain-suffix-search

22.2 Static Mappings Inside DHCP Pools

While ISC dhcpd will allow a static mapping to be defined inside the DHCP range/pool, it can result in unexpected behavior.

ISC dhcpd only checks via ping to ensure that an IP is not actively in use when making assignments. Making a static mapping does not "reserve" that IP out of the pool. The static mapping in this case merely represents a preference for an IP and others are not prevented from taking the IP if it is not in use.

An example: If the DHCP pool is from 192.168.0.10 to 192.168.0.250, and a static mapping is defined for 192.168.0.25. If the PC that normally has 192.168.0.25 is ever offline another device could be assigned 192.168.0.25. When the other machine powers back up it will not be able to get 192.168.0.25 because it is currently in use.

As such, it is best to only make assignments outside the range/pool, and the pfSense® webGUI enforces this practice.

If assignments absolutely must be made inside the pool, and the risks involved are worth taking and want to do so anyway, the input validation check may be removed from the PHP file that drives the DHCP editor page. The details of this unsupported change are left out as an exercise for the reader.

See also:

- *Status*
- *Viewing DHCPv6 Leases*
- *DHCP Logs*
- *Troubleshooting DHCPv6 Client XID Mismatches*
- *Troubleshooting Offline DHCP Leases*

CHAPTER

TWENTYTHREE

DNS

DNS, or Domain Name System, is the mechanism by which a network device resolves a name like `www.example.com` to an IP address such as `198.51.100.25`, or vice versa. Clients must have functional DNS if they are to reach other devices such as servers using their hostnames or fully qualified domain names.

23.1 DNS Resolver/Forwarder

These topics cover using pfSense® software as a caching DNS resolver or forwarder, which handles DNS requests from local clients. When acting as a resolver or forwarder, pfSense software will performs DNS resolution or hand off queries to an upstream DNS forwarding server.

23.1.1 DNS Rebinding Protections

pfSense® software includes some built in methods of protection against DNS rebinding attacks. These measures are described below.

DNS forwarder

The DNS forwarder (dnsmasq) uses the option –stop-dns-rebind by default, which rejects and logs addresses from upstream nameservers which are in the private IP ranges. In the most common usage, this is filtering DNS responses received from the Internet to prevent DNS rebinding attacks. Internet DNS responses should never come back with a private IP, hence it's safest to block this.

There are some cases when public DNS servers have private IP address replies by default, though it is not recommended. In those cases, DNS rebinding can be disabled or an override may be placed in the DNS Forwarder Advanced Settings box as follows:

```
rebind-domain-ok=/mydomain.com/
```

Note this is automatically overridden for domains in the DNS forwarder's domain override list, as the most common usage of that functionality is to resolve internal DNS hostnames.

DNS Resolver (Unbound)

Unbound has similar protections to dnsmasq, using its "Private Address support" option. With that option enabled RFC1918 addresses are stripped away from DNS answers. Additionally, the DNSSEC validator may mark the answers bogus.

In the package on 2.1 and earlier this option is located in the main "Unbound DNS Settings" tab. On 2.2 where Unbound is integrated into the base system, it is active by default and controlled by the DNS Rebinding option under System > Advanced.

Individual domains can be excluded from DNS rebinding protection using the Custom Options on the Unbound general settings. Enter one domain per line in the following format, preceded by the "server:" line.

```
server:
private-domain: "example.com"
```

Web interface protection

For those not using the DNS forwarder, and as an additional layer of checks, the web interface will block attempts to access it via an unknown hostname. It will display "Potential DNS Rebind Attack Detected" and drop any request. By default, only the hostname and domain configured under System>General Setup are accepted. For instance if firewall.example.com is configured as the system's hostname, and it is loaded in a browser using fw1.example.com, that attempt will be rejected. Additional hostnames can be added under System>Advanced, "Alternate Hostnames".

Logging in using the IP address of the system rather than the hostname does work if this message is encountered when attempting to load by hostname. Once access has been obtained, configure the hostname(s) accordingly and then it is possible to log in using the desired hostname.

If this message is encountered when a client attempted to access a forwarded service (Port forward, 1:1 NAT, relayd, etc) it indicates that the request did not match any NAT rules. From the inside of the network, this would require NAT reflection or split DNS to accomplish.

See also:

Accessing Port Forwards from Local Networks

23.1.2 Creating Wildcard Records in DNS Forwarder/Resolver

A wildcard DNS record resolves **anything**.example.com to a single IP, which can be useful in certain cases.

DNS Forwarder (dnsmasq)

A wildcard entry may be created in the DNS Forwarder by using the advanced options box, and an entry like so:

```
address=/example.com/192.168.1.54
```

That would make any host under *example.com* resolve to 192.168.1.54 (www.example.com, thissitedoesnotexist.example.com, mystuff.example.com, and so on).

If a specific Host Overrides is set for example:

```
specific.example.com 192.168.1.100
knownhost.example.com 192.168.1.101
```

Then those would be returned when doing a query for those hosts, only when no specific host has been specified in the host overrides would the advanced wildcard entry be used.

To resolve the domain to an IP address:

```
example.com 192.168.1.45
```

Leave the host field blank in the host overrides. So if the query is now for *example.com*, 192.168.1.45 would be returned. If *knownhost.example.com* was queried for then 192.168.1.101 would be returned.

If a blank hostname *example.com* host override entry has not been created, then a query for *example.com* would return the wildcard IP address set in the advanced option.

If *madeupname.example.com* was queried, then since no specific host record for *madeupname* exists in the host overrides. The wildcard entry of 192.168.1.54 would be returned.

DNS Resolver (Unbound)

The same effect may be obtained in the DNS Resolver (Unbound) using its advanced options:

```
server:
local-zone: "example.com" redirect
local-data: "example.com 86400 IN A 192.168.1.54"
```

See also:

- *DNS Lookup*
- *Troubleshooting the DNS Forwarder*

23.2 DNS Guides

How to perform various tasks related to DNS.

- *Blocking External Client DNS Queries*
- *Redirecting Client DNS Requests*
- *Troubleshooting the DNS Cache*

23.3 Dynamic DNS

Dynamic DNS updates an external DNS server with an interface IP address when it changes. This enables a firewall with a dynamic WAN such as DHCP or PPPoE to host public services even when its IP address changes periodically.

CHAPTER

TWENTYFOUR

TRAFFIC SHAPER

24.1 What the Traffic Shaper can do for a Network

The basic idea of traffic shaping is raising and lowering the priorities of packets or keeping them under a certain speed. This concept seems simple, however, the number of ways in which this concept can be applied is vast. These are but a few common examples that have proven popular with users of pfSense® software.

24.1.1 Keep Browsing Smooth

Asymmetric links, where the download speed differs from the upload speed, are commonplace, especially with DSL. Some links are so out of balance that the maximum download speed is almost unattainable because it is difficult for a firewall to send out enough ACK (acknowledgement) packets to keep traffic flowing. ACK packets are transmitted back to the sender by the receiving host to indicate that data was successfully received, and to signal that it is OK to send more. If the sender does not receive ACKs in a timely manner, congestion control mechanisms in TCP will kick in and slow down the connection.

This type of situation is common: When uploading a file over a link that has asymmetric throughput capability, browsing and downloading slows to a crawl or stalls. This happens because the uploading portion of the circuit is full from the file upload and there is little room to send ACK packets which allow downloads keep flowing. By using the shaper to prioritize ACK packets, the firewall can enable faster, more stable download speeds on asymmetric links.

This is not as important on symmetric links where the upload and download speed are the same, but may still be desirable if the available outgoing bandwidth is heavily utilized.

24.1.2 Keep VoIP Calls Clear

If Voice over IP calls use the same circuit as data, then uploads and downloads may degrade call quality. pfSense software can prioritize the call traffic above other protocols, and ensure that the calls make it through clearly without breaking up, even while streaming hi-def video from Netflix at the same time. Instead of the call breaking up, the shaper reduces speed of the other transfers to leave room for the calls.

24.1.3 Reduce Gaming Lag

The shaper also has options to give priority to the traffic associated with network gaming. Similar to prioritizing VoIP calls, the effect is that even if users on the network are downloading while playing, the response time of the game should still be nearly as fast as if the rest of the connection were idle.

24.1.4 Keep P2P Applications In Check

By lowering the priority of traffic associated with known peer-to-peer ports, administrators can rest easier knowing that even if those programs are in use, they won't hinder other traffic on the network. Due to its lower priority, other protocols will be favored over P2P traffic, which will be limited when any other services need the bandwidth.

24.1.5 Enforce Bandwidth Limits

Limiters can apply a bandwidth limit to a group of devices, such as all traffic on an interface, or masking on limiters can apply them on a per-IP address or per-network basis. This way the firewall can ensure that no one person can consume all available bandwidth.

24.2 Hardware Limitations

Traffic shaping is performed with the help of **ALTQ**. Unfortunately, only a subset of all supported network cards are capable of using these features because the drivers must be altered to support ALTQ shaping. The following network cards are capable of using traffic shaping:

> `ae(4)`, `age(4)`, `alc(4)`, `ale(4)`, `an(4)`, `aue(4)`, `axe(4)`, `bce(4)`, `bfe(4)`, `bge(4)`, `bridge(4)`, `cas(4)`, `cpsw(4)`, `cxl(4)`, `dc(4)`, `de(4)`, `ed(4)`, `em(4)`, `ep(4)`, `epair(4)`, `et(4)`, `fxp(4)`, `gem(4)`, `hme(4)`, `hn(4)`, `igb(4)`, `ix(4)`, `jme(4)`, `l2tp(4)`, `le(4)`, `lem(4)`, `msk(4)`, `mxge(4)`, `my(4)`, `ndis(4)`, `nfe(4)`, `ng(4)`, `nge(4)`, `npe(4)`, `nve(4)`, `ovpnc(4)`, `ovpns(4)`, `ppp(4)`, `pppoe(4)`, `pptp(4)`, `re(4)`, `rl(4)`, `sf(4)`, `sge(4)`, `sis(4)`, `sk(4)`, `ste(4)`, `stge(4)`, `ti(4)`, `tun(4)`, `txp(4)`, `udav(4)`, `ural(4)`, `vge(4)`, `vlan(4)`, `vmx(4)`, `vr(4)`, `vte(4)`, `vtnet(4)`, `xl(4)`

Limiters use a different backend system, operating through `dummynet` pipes in `ipfw` and not through **ALTQ**. As such, all network cards may be used for Limiters, there are no restrictions. If a firewall contains a card that does not support **ALTQ**, it may use limiters instead.

24.3 Network Interface Drivers with ALTQ Traffic Shaping Support

The intention of this page is to provide information regarding FreeBSD's ALTQ drivers, what they do, and how they work.

24.3.1 Information

The ALTQ framework is used for queuing/traffic shaping. In pfSense® software, this is utilized by the Shaper Wizard and the Queues/Interfaces tabs under **Firewall > Traffic Shaper**.

See the altq(4) or the altq(9).

On that page, select the *version of FreeBSD that corresponds to the pfSense version being run*.

In addition to the drivers listed as supporting ALTQ in FreeBSD, pfSense software also includes support for ALTQ on vlan(4) and IPsec enc(4) interfaces.

If the NIC being used does not support ALTQ, *Limiters* may be used instead.

24.4 ALTQ Scheduler Types

pfSense® software contains several ALTQ scheduler types to cover a large range of shaping scenarios. The options for ALTQ are:

Priority Queuing (PRIQ) Manages prioritization of connections

Class-Based Queuing (CBQ) Supports bandwidth sharing between queues and bandwidth limits

Hierarchical Fair Service Curve (HFSC) Supports real-time bandwidth guarantees along with a hierarchical tree of nested queues.

Controlled Delay (CoDel) Attempts to combat bufferbloat.

Fair Queuing (FAIRQ) Attempts to fairly distribute bandwidth among all connections.

PRIQ, CBQ, and HFSC are selectable in the shaper wizards and the wizards will show the proper options and create the queues based on the chosen ALTQ discipline.

24.4.1 Performance Caveats

Enabling ALTQ traffic shaping places an extra burden on the hardware, and there will be an overall potential network performance loss. On systems that have horsepower to spare, this may not be noticeable. On systems that operate close to their specification limits the firewall may see a degradation of performance. Whether the loss is worse than working without shaping depends on the individual workload.

24.4.2 Priority Queuing (PRIQ)

PRIQ is one of the easiest disciplines to configure and understand. The queues are all directly under the root queue, there is no structure to have queues under other queues with PRIQ as there is with HFSC and CBQ. It does not care about bandwidth on interfaces, only the priority of the queues. The values for priority go from `0` to `15`, and the higher the priority number, the more likely the queue is to have its packets processed.

PRIQ can be harsh to lesser queues, starving them when the higher priority queues need the bandwidth. In extreme cases, it is possible for a lower priority queue to have little or no packets handled if the higher priority queues are consuming all available resources.

24.4.3 Hierarchical Fair Service Curve (HFSC)

The HFSC traffic shaping discipline is very powerful. It is useful for services such as VoIP and video to deliver a minimum guaranteed amount of bandwidth.

Queues in HFSC are arranged in a hierarchy, or a tree, with root queues for each interface, parent queues underneath, and child queues nested under the parent queues (etc.). Each queue can have a set bandwidth and related options.

HFSC-specific Queue Options

HFSC supports a few queue options that are not supported by other disciplines. It is through these options that it achieves guaranteed real-time processing and link sharing.

The Service Curve (sc) is where bandwidth requirements for this queue are tuned.

m1 Burstable bandwidth limit

d Time limit for bandwidth burst, specified in milliseconds. (e.g. 1000 = 1 second)

m2 Normal bandwidth limit

For example, a connection needs **m1** bandwidth within **d** time, but a normal maximum of **m2**. Within the initial time set by **d**, **m2** is not checked, only **m1**. After **d** has expired, if the traffic is still above **m2**, it will be shaped. Most commonly, **m1** and **d** are left blank, so that only **m2** is checked.

Each of these values may be set for the following uses:

Upper Limit Maximum bandwidth allowed for the queue. Will do hard bandwidth limiting. The **m1** parameter here can also be used to limit bursting. In the time frame **d** a connection will not get more than **m1** bandwidth.

Real Time Minimum bandwidth guarantee for the queue. This is only valid for child queues. The **m1** parameter will always be satisfied in time frame **d**, and **m2** is the maximum that this discipline will allow to be used. Note The value for **m2** cannot exceed 30% of the available bandwidth from the parent queue.

Link Share The bandwidth share of a backlogged queue. Will share bandwidth between classes if the **Real Time** guarantees have been satisfied. The **m2** value for **Link Share** will override the **Bandwidth** setting for the queue. These two settings are the same, but if both are set, **m2** from **Link Share** is used.

By combining these factors, a queue will get the bandwidth specified by the **Real Time** factors, plus those from **Link Share**, up to a maximum of **Upper Limit**. It can take a lot of trial and error, and perhaps a lot of arithmetic, but it may be worth it to ensure that network traffic is governed properly. For more information on **m1**, **d**, and **m2** values for different scenarios, visit the pfSense Traffic Shaping forum.

24.4.4 Class-Based Queuing (CBQ)

Class-Based Queuing, or CBQ, is similar to HFSC in that is can have a tree of queues nested under other queues. It supports bandwidth limits (not guarantees like HFSC), priorities for queues, and it has the ability to allow queues to borrow bandwidth from their parent. Because of the simpler queue configuration, it can be a good alternative to HFSC especially if the firewall does not need to guarantee minimum bandwidths.

With CBQ, queue priorities range from `0` to `7` with higher numbers indicating higher priority. Queues of an equal priority are processed in a round-robin fashion.

Note: Though child queues can borrow from their parent queue, the sum of the bandwidth of the child queues cannot exceed the bandwidth of the parent. Therefore, CBQ is not an alternative to limiters for individual (e.g. per-IP address) bandwidth limits.

CBQ-Specific Queue Options

The CBQ discipline supports the concept of *borrow*, meaning that if the **Borrow from other queues when available** checkbox on the queue is enabled, then the queue will be able to borrow other available bandwidth from its parent queue. This will only allow a child queue to obtain up to the bandwidth of its *immediate* parent, if available, it will not borrow from other parent queues.

24.4.5 CoDel Active Queue Management

The CoDel Active Queue Management (AQM) discipline is short for Controlled Delay and is pronounced "coddle". It was designed to combat problems associated with bufferbloat in networking infrastructure. Bufferbloat is described in detail at http://www.bufferbloat.net/projects/bloat/wiki/Introduction. Put simply, traffic can pile up and go in chunks rather than a smooth stream due to the size of buffers in network equipment. By controlling the delay of the traffic this effect can be lessened.

CoDel has no specific configuration controls or options. When activated for a queue, it will automatically attempt to manage traffic as described in the CoDel wiki at http://www.bufferbloat.net/projects/codel/wiki. It attempts to keep traffic delays low but does permit bursting, it controls delays but it does not pay attention to round-trip delay, load, or link speed, and it can automatically adjust if the link speed changes.

The target for CoDel is mid-range networking. It does not work well at very low bandwidth (1Mbit/s or less) and it does not gracefully handle large numbers of simultaneous flows or datacenter-grade traffic loads.

CoDel is not configurable using the wizard, but it does not require complex setup:

- Navigate to **Firewall > Traffic Shaper**, **By Interface** tab
- Select an interface (e.g. **WAN**)
- Set the **Scheduler Type** to *CODEL*
- Set an appropriate value for **Bandwidth**
- Click **Save**
- Repeat as needed for all other active WAN-type interface(s)

24.4.6 Fair Queuing (FAIRQ)

In FAIRQ, queues are monitored from highest to lowest priority, but the scheduler attempts to fairly distribute bandwidth among all connections.

When there is no contention for bandwidth, FAIRQ will send all waiting packets. When there is contention for bandwidth FAIRQ will start looking for queues that are not exceeding their limits, first starting with high priority queues and working toward lower queues. A packet in a full high priority queue is processed *after* a packet from a lower priority queue which is not full. If all queues are full, then FAIRQ will send a packet from the highest priority queue.

FAIRQ allows connections to exceed queue bandwidth, but will maintain an average consumption equal to the defined queue bandwidth.

FAIRQ is not currently supported in the traffic shaper wizard and it requires a manual configuration.

24.5 Advanced Customization

The rules and queues generated by the shaper wizard may not be an exact fit for a network. Network devices may use services that need shaped which are not listed in the wizard, games that use different ports, or other protocols that need limiting.

After the basic rules have been created by the wizard, it is relatively easy to edit or copy those rules to make adjustments for other protocols.

24.5.1 Editing Shaper Queues

Queues are where bandwidth and priorities are allocated by the shaper. Each queue has settings specific to the scheduler that was chosen in the wizard (*ALTQ Scheduler Types*). Queues can also be assigned other attributes that control how they behave. Queues may be managed at **Firewall > Traffic Shaper**. Click on a queue name in the list or tree shown on the **By Interface** or **By Queue** tabs, as seen in Figure *Traffic Shaper Queues List*

> **Warning:** Creating or editing queues is for advanced users only. It is a complex task with powerful results, but without thorough understanding of the settings involved the best practice is to stick with queues generated by the wizard rather than trying to make new queues.

To edit a queue, click its name in the list/tree.

To delete a queue, click it once to edit the queue, then click  **Delete This Queue**. Do not delete a queue if it is still being referenced by a firewall rule.

To add a new queue, click the interface or parent queue under which the new queue will be placed, and then click  **Add New Queue**.

When editing a queue, each of the options must be carefully considered. For more information about these settings than is mentioned here, visit the PF Packet Queuing and Prioritization FAQ or read *The OpenBSD PF Packet Filter* book.

Name The queue name must be between 1-15 characters and cannot contain spaces. The most common convention is to start the name of a queue with the letter "q" so that it may be more readily identified in the ruleset.

Priority The priority of the queue. Can be any number from 0-7 for CBQ and 0-15 for PRIQ. Though HFSC can support priorities, the current code does not honor them when performing shaping. Queues with higher numbers are preferred by the shaper when there is an overload, so situate queues accordingly. For example, VoIP traffic is the highest priority, so it would be set to a `7` on CBQ or `15` on PRIQ. Peer-to-peer network traffic, which can be delayed in favor of other protocols, would be set at `1`.

Bandwidth (root queues) The amount of bandwidth available on this interface in the **outbound** direction. For example, WAN-type interface root queues list upload speed. LAN-type interfaces list the sum total of all WAN interface download bandwidth.

Queue Limit The number of packets that can be held in a queue waiting to be transmitted by the shaper. The default size is `50`.

Scheduler Options There are five different Scheduler Options that may be set for a given queue:

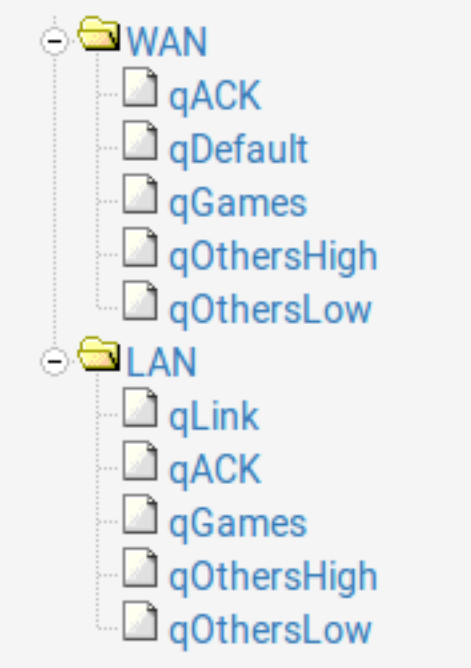


Fig. 1: Traffic Shaper Queues List

Default Queue Selects this queue as the default, the one which will handle all unmatched packets on an interface. Each interface must have one and only one default queue.

Random Early Detection (RED) A method to avoid congestion on a link. When set, the shaper will actively attempt to ensure that the queue does not get full. If the bandwidth is above the maximum given for the queue, drops will occur. Also, drops may occur if the average queue size approaches the maximum. Dropped packets are chosen at random, so connections using more bandwidth are more likely to see drops. The net effect is that the bandwidth is limited in a fair way, encouraging a balance. RED should only be used with TCP connections since TCP is capable of handling lost packets, and hosts can resend TCP packets when needed.

Random Early Detection In and Out (RIO) Enables RED with in/out, which results in having queue averages being maintained and checked against incoming and outgoing packets.

Explicit Congestion Notification (ECN) Along with RED, it allows sending of control messages that will throttle connections if both ends support ECN. Instead of dropping the packets as RED will normally do, it will set a flag in the packet indicating network congestion. If the other side sees and obeys the flag, the speed of the ongoing transfer will be reduced.

Codel Active Queue A flag to mark this queue as being the active queue for the Codel shaper discipline.

Description Optional text describing the purpose of the queue.

Bandwidth (Service Curve/Scheduler) The Bandwidth setting should be a fraction of the available bandwidth in the parent queue, but it must also be set with an awareness of the other neighboring queues. When using percentages, the total of all queues under a given parent cannot exceed 100%. When using absolute limits, the totals cannot exceed the bandwidth available in the parent queue.

Scheduler-specific Options Next are scheduler-specific options. They change depending on whether a queue is using HFSC, CBQ, or PRIQ. They are all described in *ALTQ Scheduler Types*.

Click **Save** to save the queue settings and return to the queue list, then click **Apply Changes** to reload the queues and activate the changes.

24.5.2 Editing Shaper Rules

Traffic shaping rules control how traffic is assigned into queues. If a new connection matches a traffic shaper rule, the firewall will assign packets for that connection into the queue specified by that rule.

Packet matching is handled by firewall rules, notably on the **Floating** tab. To edit the shaper rules:

- Navigate to **Firewall > Rules**
- Click the **Floating** Tab
- Find the rule to edit in the list, as shown in Figure *Traffic Shaper Rules List*
- Click  to edit an existing rule or  to create a copy of a rule
- Make any required adjustments to match different connections
- Save and Apply Changes as usual when editing firewall rules

Queues may be applied using *pass* rules on interface tabs, but the wizard only creates rules on the **Floating** tab using the *match* action that does not affect whether or not a connection is passed or blocked; it only queues traffic. Because these rules operate the same as any other rules, any criteria used to match connections may be used to queue.

See also:

For more information on floating rules, see *Floating Rules* and *Configuring firewall rules* for information on firewall rules in general.

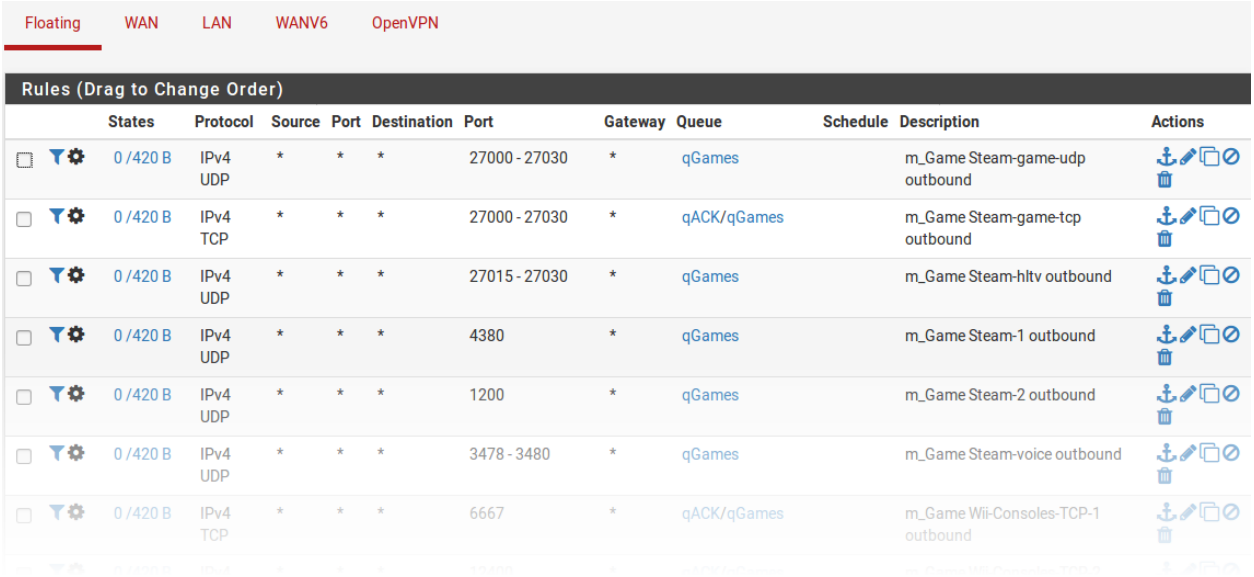


Fig. 2: Traffic Shaper Rules List

Shaper Rule Matching Tips

Connections can be tricky to match properly due to several factors, including:

- NAT applies before outbound firewall rules can match connections, so for connections that have outbound NAT applies as they leave a WAN-type interface, the private IP address source is hidden by NAT and cannot be matched by a rule.
- Some protocols such as Bittorrent will use random ports or the same ports as other services.
- Multiple protocols using the same port cannot be distinguished by the firewall.
- A protocol may use a range of ports so wide that it cannot be distinguished from other traffic.

While many of these cannot be solved by the firewall directly, there are ways to work around these limitations in a few cases.

To match by a private address source outbound in WAN floating rules, first tag the traffic as it passes in on a local interface. For example, match inbound on LAN and use the advanced **Tag** field to set a value, and then use the **Tagged** field on the WAN-side floating rule to match the same connection as it exits the firewall. Alternately, queue the traffic as it enters the LAN with a pass rule instead of when it exits a WAN.

Match by address instead of port/protocol where possible to sort out ambiguous protocols. In these cases, either the local source or the remote destination may be a single address or a small set of addresses. For example, matching VoIP traffic is much simpler if the firewall can match the remote SIP trunk or PBX rather than attempting to match a wide range of ports for RTP (e.g. 10000- 20000).

If bittorrent is allowed on a network but must be shaped, then dedicate a specific local device that is allowed to use bittorrent and then shape all connections to/from that device as Peer-to-Peer traffic.

24.5.3 Removing Traffic Shaper Settings

To remove all traffic shaper queues and rules created by the wizard:

- Navigate to **Firewall > Traffic Shaper**
- Click the **By Interface** tab
- Click  **Remove Shaper**
- Click **OK** on the confirmation prompt

24.6 Traffic Shaping with Differentiated Services (DiffServ) Identifiers

pfSense® software supports **Differentiated services** (**DiffServ**) for traffic filtering or queue assignments. DiffServ takes the place of the outdated **Type of service** (**TOS**). DiffServ uses the upper six bits of the TOS field in the IP header (the six bits being called the *DiffServ Code Point field*), while the lower two bits are reserved for Explicit Congestion Notification (ECN).

Unless appropriately configured, pfSense software ignores the content of the DiffServ Code Point (DSCP) field. To prioritize traffic, the *Traffic Shaper* needs to be set up accordingly.

Warning: pfSense software *does not* support the setting or changing of DiffServ values, only matching.

24.6.1 Supported DiffServ Code Point Values

Note that the interpretations of the DSCP values, as provided by the various RFCs, are only given as a reference. How the DSCP values are interpreted in any specific setup is entirely up to the user or end nodes.

The **Assured Forwarding** (AF) Behavior Group is recommended in RFC 2597.

Table 1: Assured Forwarding (AF) Behavior Group values

Precedence	Class 1 (lowest)	Class 2	Class 3	Class 4 (highest)
Low Drop	AF11 (10/0x0a)	AF21 (18/0x12)	AF31 (26/0x1a)	AF41 (34/0x22)
Med Drop	AF12 (12/0x0c)	AF22 (20/0x14)	AF32 (28/0x1c)	AF42 (36/0x24)
High Drop	AF13 (14/0x0e)	AF23 (22/0x16)	AF33 (30/0x1e)	AF43 (38/0x26)

For low-drop/low-latency traffic, use EF and VA DSCP values.

Table 2: Expedited Forwarding (EF) and Voice Admit (VA) values

PHB	DSCP Value	RFC
Expedited Forwarding (EF)	46/0x2e	RFC 3246
Voice Admit (VA)	44/0x2c	RFC 5865

The Class Selector (CS) PHB group has been retained from TOS.

Table 3: Class Selector (CS) values

Class Selector	DSCP Value
CS1	8/0x08
CS2	16/0x10
CS3	24/0x18
CS4	32/0x20
CS5	40/0x28
CS6	48/0x30
CS7	56/0x38

To provide limited backward comparability to TOS, pfSense also recognizes the following DSCP/TOS values.

Table 4: TOS Compatibility values

TOS	DSCP Value	TOS value
reliability	1/0x01	4/0x04
throughput	2/0x02	8/0x08
lowdelay	4/0x04	16/0x10

pfSense only matches exact values. All six bit in the DSCP field must match.

24.6.2 Caveats

By default, pfSense matches only the first packet of a connection, which is the packet that creates an entry in the state table. If a connection starts with a different DSCP value, has no DSCP value in the starting packet, or otherwise changes DSCP values during the connection, the traffic will not be classified as expected.

Tip: This can be worked around by using "no state" rules, but crafting these rules in a secure manner is difficult, so it is not a workaround that we recommend.

24.6.3 Adding additional DSCP values for experimental use

Assuming basic knowledge about PHP, it is possible to add additional DiffServ Code Point values by editing `/usr/local/www/guiconfig.inc`. In this file, the variable `$firewall_rules_dscp_types` is initialized with an array containing the recognized DSCP values. New values can be specified as hex values, optionally followed by a blank and a comment like, for example:

```
"0x03",
```

Valid values are in the range `0x01` through `0x3f`.

Caution: These changes will be lost upon a firmware update.

24.6.4 RFCs

- RFC 2474 — Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
- RFC 2475 — An Architecture for Differentiated Services
- RFC 2597 — Assured Forwarding PHB Group
- RFC 2983 — Differentiated Services and Tunnels
- RFC 3086 — Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification
- RFC 3140 — Per Hop Behavior Identification Codes (replaces RFC 2836)
- RFC 3246 — An Expedited Forwarding PHB (Per-Hop Behavior) (obsoletes RFC 2598)
- RFC 3247 — Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior)
- RFC 3260 — New Terminology and Clarifications for Diffserv (updates
- RFC 2474, RFC 2475 and RFC 2597)
- RFC 4594 — Configuration Guidelines for DiffServ Service Classes
- RFC 5865 — A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic (updates RFC 4542 and RFC 4594)
- RFC 3289 — Management Information Base for the Differentiated Services Architecture
- RFC 3290 — An Informal Management Model for Diffserv Routers
- RFC 3317 — Differentiated Services Quality of Service Policy Information Base

24.7 Limiters

Limiters are an alternate method of traffic shaping. Limiters use dummynet(4) to enact bandwidth limits and perform other prioritization tasks, and they do not rely on ALTQ. Limiters are currently the only way to achieve per-IP address or per-network bandwidth rate limiting using pfSense® software. Limiters are also used internally by Captive Portal for per-user bandwidth limits.

Limiters are managed at **Firewall > Traffic Shaper** on the **Limiters** tab.

Like HFSC and CBQ, Limiters may be nested with queues inside other queues. Root-level limiters (Also called Pipes), may have bandwidth limits and delays, while child limiters (Also called queues), may have priorities (Also called weights). Bandwidth limits can be optionally masked by either the source or destination IP address, so that the limits can be applied on a per-IP address or network basis instead of as a general group.

Limiters are nearly always used in pairs: One for incoming traffic and one for outgoing traffic.

According to its man page the dummynet(4) system was originally designed as a means to test TCP congestion control and it grew up from there. Due to this purpose, a unique feature of limiters is that they can be used to induce artificial packet loss and delay into network traffic. That is primarily used in troubleshooting and testing (or being evil and playing a prank on someone), and not often found in production.

24.7.1 Uses for Limiters

The primary use for limiters is to apply bandwidth limits for users or specific protocols, e.g. “Maximum of 1Mbit/s for SMTP”, or “Joe’s PC only can use 5Mbit/s”. Limiters can apply a per-IP address or per-network limit, such as “All Users in 192.168.50.0/24 can use a maximum of 3Mbit/s each” or “The guest network and public network can use 1Mbit/s for each segment”.

Limiters are the only type of shaper available in pfSense software which is capable of oversubscription in this manner. The ALTQ shaper requires all child queues to sum up to no more than the speed of the parent queue, but masked limiters allow a set limit to as many IP addresses as can be funneled through the limiter by firewall rules.

Conceptually, consider a limiter as a bucket of bandwidth. All traffic flowing through an unmasked limiter draws bandwidth from the same bucket. Masking a limiter effectively sets up multiple buckets of the same size, one per masked group. Whether that is a single host or an entire network depends on the mask value.

Limiters can also allow for reserved bandwidth by limiting everything *except* a specific protocol which can then consume all remaining bandwidth. In this type of setup on a 10Mbit/s link the firewall would pass traffic from, for example, a SIP server with no limiter. Then the firewall would use a pass rule for all other traffic with a limit of 8Mbit/s. This would let the SIP server use all of the bandwidth it wanted, but it would always have a minimum of 2Mbit/s to itself.

24.7.2 How Limiters Work

Limiters, like ALTQ, hold traffic to a certain point by dropping or delaying packets to achieve a specific line rate. Usually taking advantage of built-in mechanisms from protocols that detect the loss and back off to a sustainable speed.

In situations where packets are queued under the same parent pipe, the firewall considers their weights when ordering the packets before it sends them. Unlike priorities in CBQ and PRIQ, the weight of a queue in a limiter will never starve it for bandwidth.

24.7.3 Limiters and IPv6

Limiters work with IPv6, though it requires separate IPv4 and IPv6 rules to apply limiters properly.

24.7.4 Limitations

Limiter pipes do not have a concept of borrowing bandwidth from other pipes. A limit is always a hard upper limit.

Limiters use IPFW, so there will be additional (though small) overhead from the IPFW kernel module and the extra packet processing involved.

Limiters cannot effectively guarantee a minimum bandwidth amount for a pipe or queue, only a maximum.

Child queues cannot have bandwidth values, so a pipe cannot be split into smaller pipes by queues. Child queues can only use weights to prioritize packets inside a pipe.

The overhead from delaying and queuing packets can cause increased mbuf usage. For more information on increasing the amount of available mbufs, see *Hardware Tuning and Troubleshooting*.

Limiters and Multi-WAN

When using limiters with Multi-WAN, limits for non-default gateways must be applied using floating rules set for the *out* direction and configured with the appropriate gateway.

24.7.5 Creating Limiters

Limiters are managed under **Firewall > Traffic Shaper** on the **Limiters** tab.

To create a new root-level limiter (pipe), click  **New Limiter**.

To create a child limiter (queue), click an existing limiter under which it can be created, and click  **Add New Queue**.

Tip: In nearly all cases, limiters exist in pairs at the same level (e.g. two pipes, or two queues): One for inbound traffic and one for outbound traffic. When creating new limiters or queues, create one for each direction.

Enable Check the box to enable this limiter. If the limiter is disabled, it will not be available for use by firewall rules.

Name This defines the name of the limiter, as it will appear for selection on firewall rules. The name must be alphanumeric, and may also include – and _.

Tip: When choosing a name, avoid using `In` and `Out` since the same limiter, if used on both WAN and LAN, would be used in the *In* direction on one interface and the *Out* direction on another. The best practice is to use `Down` or `Download` and `Up` or `Upload`.

Bandwidth (Pipes) This section defines a bandwidth value for the pipe, or multiple bandwidths if schedules are involved. This option does not appear when editing a child limiter (queue).

Bandwidth The numerical part of the bandwidth for the pipe, e.g. `3` or `500`.

Bw Type The units for the **Bandwidth** field, such as *Mbit/s*, *Kbit/s*, or *Bit/s*.

Schedule If the firewall has schedules defined (*Time Based Rules*), the firewall offers them in this list. When schedules are in use by the firewall, the limiter can have a bandwidth value for each potential schedule. Define these by clicking  **Add Schedule** to add another bandwidth definition.

If a limiter contains multiple bandwidth specifications, they must each use a different schedule. For example if the firewall has a "Work Day" schedule, then it must also have an "Off Hours" schedule that contains all of the time not included in "Work Day" for the second bandwidth specification.

Mask This drop-down list controls how the limiter will mask addresses in the pipe or queue.

None When set to *none*, the limiter does not perform any masking. The pipe bandwidth will be applied to all traffic as a whole.

Source / Destination address When a limiter is set for *Source Address* or *Destination Address*, the pipe bandwidth limit will be applied on a per-IP address basis or a subnet basis, depending on the masking bits, using the direction chosen in the masking.

In general, a limiter should mask the **Source Address** on **Upload** (In) limiters for LAN-type interfaces, and **Destination Address** on **Download** (Out) limiters on LAN-type interfaces. Similar to swapping the directionality of the limiters when applying to LAN and WAN, masking is swapped as well, so the same masked limiter set for **In** on LAN should be used for **Out** on WAN.

Mask Bits There are separate boxes to control the address masking for IPv4 and IPv6. For IPv4 a value of *32* for **IPv4 mask bits** sets up a per-IPv4 address limit, which is the most common usage. For a per-IPv6-address limit, use *128* as the **IPv6 mask bits** value.

To create per-subnet or similar masks, enter the subnet bits in the appropriate field for either IPv4 or IPv6 mask bits, such as *24* to limit IPv4 in groups of /24 subnets.

Description An optional bit of text to explain the purpose for this Limiter.

Advanced Options Additional options that vary when editing a pipe or a queue.

Delay (Pipes) The **Delay** option is only found on limiter pipes. It introduces an artificial delay (latency), specified in milliseconds, into the transmission of any packets in the limiter pipe. This is typically left blank so that packets are transmitted as fast as possible by the firewall. This can be used to simulate high-latency connections such as satellite uplinks for lab testing.

Weight (Queues) The **Weight** option is only found on child limiters (queues). This value can range from 1 to 100. Higher values give more precedence to packets in a given queue. Unlike PRIQ and CBQ priorities, a lowly-weighted queue is not in danger of being starved of bandwidth by the firewall.

Packet loss rate Another method of artificially degrading traffic. The **Packet Loss Rate** can be configured to drop a certain fraction of packets that enter the limiter. The value is expressed as a decimal representation of a percentage, so `0.01` is 1%, or one packet out of a hundred dropped. This field is typically left empty so every packet is delivered by the firewall.

Queue Size Sets the size of the queue, specified in queue slots, used for handling queuing delay. Left blank, it defaults to `50` slots, which is the recommended value. Slow speed links may need a lower queue size to operate efficiently. High speed links may need more slots.

> **Tip:** In cases where there are several limiters or limiters with large **Queue Size** values, a **System Tunable** may need set to increase the value of `net.inet.ip.dummynet.pipe_slot_limit` above the total number of configured queue lots among all pipes and queues.

Bucket Size The **Bucket Size**, also specified in slots, sets the size of the hash table used for queue storage. The default value is `64`. It must be a numeric value between `16` and `65536`, inclusive. This value is typically left blank.

See also:

For more information about these values, consult the ipfw(8) man page, in the section titled "Traffic Shaper (Dummynet) Configuration".

24.7.6 Assigning and Using Limiters

Limiters are assigned using firewall rules via the **In/Out Pipe** selectors under **Advanced Options**. Any potential matching criteria that a firewall rule supports can assign traffic to a limiter.

The most important thing to remember when assigning a limiter to a rule is that the **In** and **Out** fields are designated **from the perspective of the firewall itself**.

For example, in a firewall configuration with a single LAN and single WAN, inbound traffic on a LAN interface is leaving toward the Internet, i.e. *uploaded* data. Outbound traffic on the LAN interface is going toward the client PC, i.e. *downloaded* data. On the WAN interface the directionality is reversed; Inbound traffic is coming from the Internet to the client (download), and outbound traffic is going from the client to the Internet (upload).

In most cases, a firewall rule will have both an **In** limiter and **Out** limiter, but only the **In** limiter is required by the firewall to limit traffic in a single direction.

Limiters may be applied on normal interface rules, or on floating rules. On floating in the *out* direction, the In/Out selections are flipped conceptually.

24.7.7 Checking Limiter Usage

Information about active limiters may be found under **Diagnostics > Limiter Info**. Here, each limiter and child queue is shown in text format.

The set bandwidth and parameters for each limiter are displayed by the page, along with the current traffic level moving inside the limiter. In the case of masked limiters, the firewall displays the bandwidth of each IP address or masked group.

24.8 Traffic Shaping and VPNs

The following discussions pertain primarily to ALTQ shaping. Limiters will work fine with VPNs as they would with any other interface and rules. Only the ALTQ shaper requires special consideration.

Traffic shaping with VPNs is a tricky topic because VPN traffic is considered separate from, but also a part of, the WAN traffic through which it also flows. If WAN is 10 Mbit/s, then the VPN can also use 10Mbit/s, but there is not actually 20Mbit/s of bandwidth to consider, only 10Mbit/s. As such, methods of shaping that focus more on prioritization than bandwidth are more reliable, such as PRIQ or in some cases, CBQ.

If all traffic inside the VPN must be prioritized by the firewall, then it is enough to consider only the VPN traffic itself directly on WAN, rather than attempting to queue traffic on the VPN separately. In these cases, use a floating rule on

WAN to match the VPN traffic itself. The exact type of traffic varies depending on the type of VPN. IPsec and PPTP traffic on WAN can both be prioritized by the shaper wizard, and these rules can be used as an example to match other protocols.

24.8.1 OpenVPN

With OpenVPN, multiple interfaces exist on the operating system, one per VPN. This can make shaping easier in some cases. Features of OpenVPN can also make it easier to shape traffic on WAN and ignore the tunnel itself.

Shaping inside the tunnel

If multiple classes of traffic are carried on the tunnel, then prioritization must be done to the traffic inside the tunnel. In order for the wizard to consider the traffic in this way, the VPN must be assigned as its own interface in the GUI. To accomplish this, assign it as described in *Interface assignment and configuration*, and then use the shaper wizard as if it were a separate WAN interface, and classify the traffic as usual.

Shaping outside the tunnel (passtos)

If the primary concern is shaping VoIP traffic over a VPN, another choice to consider is the `passtos` option in OpenVPN, called **Type-of-Service** in the OpenVPN client or server options. This option copies the TOS bit from the inner packet to the outer packet of the VPN. Thus, if the VoIP traffic has the TOS (DSCP) portion of the packet header set, then the OpenVPN packets will also have the same value.

This option is more useful for signaling intermediate routers about the QoS needs, however. Though the DSCP option on firewall rules can match based on TOS bits, as described in *Diffserv Code Point*, such matching would have to occur in the packet creating a firewall state, and not on specific packets flowing through that state.

Note: Because this option tells OpenVPN to copy data from the inner packet to the outer packet, it does expose a little information about the type of traffic crossing the VPN. Whether or not the information disclosure, though minor, is worth the risk for the gains offered by proper packet prioritization depends on the needs of the network environment.

24.8.2 IPsec

IPsec is presented to the operating system on a single interface no matter how many tunnels are configured and no matter which WANs are used by the tunnels. This makes shaping IPsec traffic difficult, especially when trying to shape traffic inside one particular IPsec tunnel.

The IPsec interface is also not possible to use on its own as an interface with the wizard. Floating rules can match and queue traffic on the IPsec interface, but in most cases only inbound traffic will be queued as expected. Actual results may vary.

24.9 Traffic Shaping UPnP Connections

UPnP rules are generated dynamically by the UPnP daemon happen outside of the typical user rules. There is a configuration option for UPnP where a queue can be defined to which UPnP will direct traffic that is directed through the rules it creates. This may be set through the pfSense® webGUI at **Services > UPnP & NAT-PMP**, and type in a valid **Traffic Shaping Queue**.

See also:

Monitoring the Queues Using the Shaper Wizard to Configure ALTQ Traffic Shaping Troubleshooting Traffic Shaping Troubleshooting Traffic Shaping Graphs

Traffic shaping, or network Quality of Service (QoS), is a means of prioritizing network traffic. Without traffic shaping, packets are processed on a first in/first out basis by the firewall. QoS offers a means of prioritizing different types of traffic, ensuring that high priority services receive the bandwidth they need before lesser priority services.

For simplicity, the traffic shaping system in pfSense® software may also be referred to as the "shaper", and the act of traffic shaping may be called "shaping".

24.10 Traffic Shaping Types

There are two types of QoS available in pfSense software: ALTQ and Limiters.

The **ALTQ** framework is handled through `pf` and is closely tied to network card drivers. ALTQ can handle several types of schedulers and queue layouts. The traffic shaper wizard configures ALTQ and gives firewall administrators the ability to quickly configure QoS for common scenarios, and it allows custom rules for more complex tasks. ALTQ is inefficient, however, so the maximum potential throughput of a firewall is lowered significantly when it is active.

pfSense software also supports a separate shaper concept called **Limiters**. Limiters enforce hard bandwidth limits for a group or on a per-IP address or network basis. Inside of those bandwidth limits, limiters can also manage traffic priorities.

24.11 Traffic Shaping Basics

For administrators who are unfamiliar with traffic shaping, it is like a bouncer at an exclusive club. The VIPs (Very Important Packets) always make it in first and without waiting. The regular packets have to wait their turn in line, and "undesirable" packets can be kept out until after the real party is over. All the while, the club is kept at capacity and never overloaded. If more VIPs come along later, regular packets may need to be tossed out to keep the place from getting too crowded.

ALTQ shaping concepts can be counter-intuitive at first because the traffic has to be queued in a place where the operating system can control the flow of packets. Incoming traffic from the Internet going to a host on the LAN (downloading) is shaped *leaving* the LAN interface from the firewall. In the same manner, traffic going from the LAN to the Internet (uploading) is shaped when leaving the WAN.

For ALTQ, there are traffic shaping queues, and traffic shaping rules. The queues allocate bandwidth and priorities. Traffic shaping rules control how traffic is assigned into those queues. Rules for the shaper work the same as firewall rules, and allow the same matching characteristics. If a packet matches a shaper rule, it will be assigned into the queues specified by that rule. In pfSense software, shaper rules are mostly handled on the **Floating** tab using the *Match* action that assigns the traffic into queues, but rules on any interface can assign traffic into queues using the *Pass* action.

Limiter rules are handled differently. Limiters apply on regular pass rules and enforce their limits on the traffic as it enters and leaves an interface. Limiters almost always exist in pairs: One for the "download" direction traffic and one for the "upload" direction traffic.

CHAPTER

TWENTYFIVE

CAPTIVE PORTAL

25.1 Captive Portal Zones

Captive Portal zones define separate portals for different sets of interfaces. For example, LAN and Wireless could use one portal, while a conference room would get a separate portal page. Each zone has separate settings for HTML pages, authentication, allowed addresses, and so on. A zone must be created before its settings can be changed.

Note: A zone may have multiple interfaces, but an interface may only be a member of one zone. Attempting to add the same interface to multiple zones will result in an error.

25.1.1 Managing Captive Portal Zones

Captive Portal zones are managed at **Services > Captive Portal**. A list of zones is displayed there, and zones may be added, edited, or deleted from that list.

To create a new Captive Portal zone:

- Navigate to **Services > Captive Portal**
- Click  **Add**
- Enter a **Zone Name**, which may only consist of letters, digits, numbers, and underscores. Spaces and other special characters may not be used
- Enter an optional **Zone Description** to further describe the zone, if desired
- Click **Save & Continue** to move on to the portal settings for the zone

To edit an existing zone, click  at the end of its row.

To delete an existing zone, click  at the end of its row, and then click to confirm the action.

25.2 Common Captive Portal Scenarios

The following are some basic, common scenarios for the use of a Captive Portal. The details of how to perform all of the actions described will be covered throughout this chapter.

25.2.1 Portal Configuration Without Authentication

For a simple portal without authentication:

- Create a new Zone
- Check **Enable captive portal**
- Select an **Interface**
- Upload an HTML page with the portal contents as described in *Portal page without authentication*
- Click **Save**

Additional configuration options may be added as detailed in *Zone Configuration Options*.

25.2.2 Portal Configuration Using Local Authentication or Vouchers

To setup a portal with local authentication:

- Create a Zone
- Check **Enable captive portal**
- Select an **Interface**
- Set **Authentication Method** to **Local User Manager / Vouchers**
- Upload an HTML page with the portal contents as described in *Portal page with authentication*.

Additional configuration options may be added as detailed in *Zone Configuration Options*. Then configure the local users in the **User Manager** (*User Management and Authentication*).

To use vouchers, proceed to the **Vouchers** tab and create them there. See *Vouchers* for more information on Vouchers, and use the sample portal page HTML code from *Portal page with Vouchers*.

25.2.3 Portal Configuration Using RADIUS Authentication

To setup a portal using RADIUS authentication:

- Configure the RADIUS server to allow requests from the firewall
- Create a Zone
- Check **Enable captive portal**
- Select an **Interface**
- Set **Authentication Method** to **RADIUS Authentication**
- Fill in the settings for **Primary RADIUS Server** under **Primary Authentication Source**

Read the next section for information on specific configuration options.

25.3 Zone Configuration Options

This section describes each of the configuration options for a Captive Portal zone. Options for a zone are independent of those for other zones. For example, allowed IP address entries in a zone only affect that specific zone.

To reach this page, navigate to **Services > Captive Portal** and edit an existing zone from the list with  , or click  **Add** to create a new zone.

Enable Check to enable this Captive Portal zone.

Description Brief text describing the purpose of the zone.

Interface Determines the interfaces that used by this Captive Portal zone. This **cannot** be a WAN interface. It can be a bridge interface so long as it is the actual bridge (e.g. `bridge0`) and the bridge interface has an IP address assigned.

Maximum concurrent connections Specifies the maximum number of concurrent connections to the portal web server per IP address. The default value is `4`, which is sufficient for most environments. This limit exists to prevent a single host from exhausting all resources on the firewall, whether inadvertent or intentional.

One example where this would otherwise be a problem is a host infected with a worm. The thousands of connections issued will cause the captive portal page to be generated repeatedly if the host is not authenticated already, which would otherwise generate so much load it could leave the firewall unresponsive.

Idle timeout A timeout, specified in minutes, after which idle users will be disconnected by the portal. Users may log back in immediately.

Hard timeout A timeout, specified in minutes, after which the portal will forcefully log off users.

Tip: Set either a hard timeout, idle timeout, or both to ensure sessions are removed by the portal when users do not log off manually.

Users may log back in immediately after the hard timeout if their credentials are still valid (for local accounts, not expired, and for RADIUS authentication, user can still successfully authenticate to RADIUS).

Note: If a timeout value is set, the timeout must be less than the DHCP lease time or captive portal sessions can remain active for IP addresses that have switched to different devices. Setting the timeout lower will ensure that the portal sessions end before the lease would be reallocated to a new client.

Traffic Quota An amount of traffic which, when exceeded by a client, will trigger a disconnect of that client by the portal. This includes both upload and download traffic. Users may log back in immediately if their credentials are still valid

Pass-Through Credits These credits give devices a grace period before they must authenticate via the portal. For example, a device could connect 3 times within a day without seeing the portal page, but any more than that and they must login. By setting the hard timeout to a value such as 1 hour, the portal would effectively limit a client to three hours of access before forcing it to authenticate. By default this is disabled, and all clients are presented with the portal login page and must login.

Note: For this to be effective, set a hard timeout and/or idle timeout.

Pass-through credits allowed per MAC address The number of times a specific MAC address may connect through the portal. Once the client uses its credits, it can only log in with valid credentials until the waiting period has expired.

Waiting period to restore pass-through credits The number of hours after which the portal will restore the pass-through credits for a client to the original count after it uses the first one. This must be above `0` hours.

Reset waiting period on attempted access If enabled, the waiting period is reset by the portal to the original duration if access is attempted when all pass-through credits have already been exhausted. This prevents people who repeatedly attempt to access the portal from gaining open access too quickly.

Logout popup window When checked, the portal attempts to show a logout pop up window to the user which allows clients to explicitly disconnect themselves before the idle or hard timeout occurs. Unfortunately, since most browsers block pop up windows, this window may not work for most clients unless.

Pre-authentication redirect URL As the name implies, this option redirects users to the specified URL *before* they authenticate. Commonly, this is used to display a custom landing page describing the device location hosted on a server locally or elsewhere. That landing page must contain a link which in turn redirects the users back to the portal page, e.g. `http://x.x.x.x:8002/index.php?zone=somezone&redirurl=http%3A%2F%2Fsomesite.example.com`.

See also:

See *Allowed Hostnames* to allow hostnames through the portal without authentication, and *Allowed IP Address* for IP addresses.

The custom captive portal page must have extra code at the top to properly handle this redirect. In the example code below, the pre-authentication redirect target page must also put its own URL in the `redirurl` parameter of its link back to the portal in order for the login page to appear.

```
<?php
require_once("globals.inc");
$request_uri = urldecode(str_replace("/index.php?zone={$_REQUEST['zone']}&redirurl=",
↪"", $_SERVER["REQUEST_URI"]));
$portal_redirurl = urldecode("$PORTAL_REDIRURL$");
if(!stristr($portal_redirurl, $request_uri)) {
      Header("Location: $PORTAL_REDIRURL$");
      exit;
}
?>
```

After authentication Redirection URL After authenticating or clicking through the portal, it will redirect users to this URL rather than the one they originally tried to access. If this field is left blank, the portal will redirect the user to the URL they initially attempted to access.

Blocked MAC address redirect URL URL to which the portal will redirect users with blocked MAC addresses when they attempt access through the portal.

Preserve user database When set, the database containing logged-in users is preserved by the portal when the firewall reboots.

Concurrent user logins Controls whether or not users are allowed to connect multiple times. This is not a total limit for the entire portal, but a per-account limit.

May be set to one of the following:

Disabled The portal will not allow concurrent logins for a user or voucher.

Multiple (Default) The portal does not enforce any restrictions on concurrent logins by a user or voucher.

Last Login The portal will only allow only one login per user account or voucher. The most recent login is permitted and any previous logins are disconnected.

First Login The portal will only allow only one login per user account or voucher. The portal permits the first login and denies any subsequent login attempt.

MAC filtering When set, the portal disables MAC address filtering. This is necessary in cases where the MAC address cannot reliably be determined, such as when multiple subnets exist behind a separate router using the portal. In that type of situation, all users behind a router will show up to the portal with the MAC address of the intermediate router. If this option is set, the portal will not attempt to ensure that the MAC address of clients stay the same while they are logged into the portal.

Note: This option is not compatible with RADIUS MAC authentication.

Pass-through MAC Auto Entry In certain use cases, users may only need to authenticate once per device, and then the client should not see the portal login again unless they change devices. Setting up pass-through MAC entries can automatically achieve this goal.

Pass-through MAC automatic additions If this option is set, the portal automatically adds a MAC passthrough entry after the user has successfully authenticated. Users of that MAC address will never have to authenticate again unless so long as the entry is present in the configuration. To remove the passthrough MAC entry, log in and remove it manually from the Pass-through MAC tab.

Note: This option is not compatible with RADIUS MAC authentication or the logout window.

Pass-through MAC automatic addition with username If this option is set, the portal saves the username used during authentication along with the pass-through MAC entry. To remove the passthrough MAC entry, log in and remove it manually from the Pass-through MAC tab.

Per-user bandwidth restrictions Captive Portal can also optionally rate-limit users to keep them from using too much bandwidth. The **Default download** and **Default upload** fields define the default values for user bandwidth, specified in Kilobits per second. These values can be overridden by RADIUS (*Passing back configuration from RADIUS Servers*) for different limits for specific users. If the fields are blank or set to `0`, then users have unlimited bandwidth.

Use Custom Captive Portal Page When set, enables upload controls for manually crafted portal pages. See *HTML Page Contents* for details. When unset, simple portal page customization controls are available (*Captive Portal Login Page*).

25.3.1 Captive Portal Login Page

These simple customization controls enable small changes to the portal page without writing custom HTML. For more complicated portal pages, see *HTML Page Contents*.

Display Custom Logo Image When set, the portal page includes the custom image from **Logo Image** instead of the default logo.

Logo Image Upload control for setting a custom logo image.

Display Custom Background Image When set, the portal page includes the custom image from **Background Image** instead of the default background.

Background Image Upload control for setting a custom background image.

Terms and Conditions Text displayed by the portal to the user to which the user must agree before they are permitted to login.

25.3.2 Authentication

This section configures authentication for Captive Portal. If authentication is required for the zone it may be handled by the local user database, RADIUS, or LDAP.

Authentication Method

Use an Authentication backend This option allows users to authenticate with a username and password or vouchers. The authentication is handled by the local user database (*User Management and Authentication*) or an authentication server (*Authentication Servers*).

Vouchers are pre-generated access codes which grant short-term access to users. Vouchers may be used in addition to, or instead of, user authentication. For more information on using vouchers, see *Vouchers* later in this section.

None, don't authenticate users The portal only requires users to click through the login page for access. The form must still be submitted, but it does not need to have any user entry fields, only a submit button.

Use RADIUS MAC Authentication The portal attempts to authenticate users by sending their MAC address as the username and the password entered into **MAC authentication secret** to the RADIUS server.

Note: Users must still attempt an HTTP connection so the portal will see the attempt and perform the initial authentication.

See *RADIUS MAC Authentication Options* for additional options.

This option is not available if MAC filtering is disabled.

Authentication Server A multi-select control where one or more primary authentication servers, or the local database, can be set for use by the portal. See *Primary Authentication Source* for more information.

Local Database Captive Portal users in this mode are managed in the pfSense® GUI. Local users are added in the User Manager (*Manage Local Users*).

Additionally, the **Local Authentication Privileges** option can limit access to only users who possess the proper access privileges.

LDAP Server When an LDAP server is active in the control, it is used by the portal for authentication as-is. There are no additional options for LDAP server behavior.

RADIUS Server When a RADIUS server is active in the control, numerous RADIUS server options are displayed by the GUI and Captive Portal users in this zone will be validated against the configured RADIUS server(s).

Secondary Authentication Server Similar to **Authentication Server**, but sets up an additional separate means of authentication using distinct fields. See *Secondary Authentication Source* for more information.

Primary Authentication Source

The Primary/Secondary authentication servers are used for the main username and password fields on the login form, `auth_user` and `auth_pass`, such as:

```
<tr>
    <td align="right">Username:</td>
    <td><input name="auth_user" type="text" style="border: 1px dashed;"></td>
 </tr>
<tr>
    <td align="right">Password:</td>
    <td><input name="auth_pass" type="password" style="border: 1px dashed;"></td>
</tr>
```

If the first server is down, the portal will attempt authentication using the other servers in the list, in order (top down).

Secondary Authentication Source

The secondary authentication source defines a completely separate authentication setup from the primary. For example, the primary source could be traditional usernames and passwords, while the secondary could be pre-paid card numbers or PINs.

The secondary authentication source uses the form fields `auth_user2` and `auth_pass2` in the captive portal HTML, such as:

```
<tr>
    <td align="right">Username:</td>
    <td><input name="auth_user2" type="text" style="border: 1px dashed;"></td>
</tr>
<tr>
    <td align="right">Password:</td>
    <td><input name="auth_pass2" type="password" style="border: 1px dashed;"></td>
</tr>
```

If the first server is down, the portal will attempt authentication using the other servers in the list, in order (top down).

Local Database Options

Local Authentication Privileges When **Allow only users/groups with 'Captive portal login' privilege set** is active, the portal will limit access to only users who have Captive Portal privilege. The privilege can be directly on their account or inherited via group membership..

RADIUS Authentication Options

RADIUS is a means of authenticating users against a central server that contains account information. There are many implementations of RADIUS, such as FreeRADIUS, Radiator, and NPS on Windows servers.

RADIUS accounting can be enabled to send usage information for each user to the RADIUS server. Refer to documentation for the RADIUS server for more information.

See also:

To add or edit RADIUS server entries on pfSense software, see *Authentication Servers*.

See also:

For those with a Microsoft Active Directory network infrastructure, RADIUS can be used to authenticate captive portal users from Active Directory using Microsoft NPS. This is described in *Authenticating from Active Directory using RADIUS/NPS*.

Passing back configuration from RADIUS Servers

Some default Captive Portal settings can be overridden by reply attributes from RADIUS servers. The exact attributes can vary by vendor, and may not be supported by all RADIUS servers.

User bandwidth restrictions Defines the bandwidth for the user, drawn from common options such as `WISPr-Bandwidth-Max-Up`/`WISPr-Bandwidth-Max-Down`, or `ChilliSpot-Bandwidth-Max-Up`/`ChilliSpot-Bandwidth-Max-Down`.

Session Timeout Drawn from the RADIUS attribute `Session-Timeout`, it will disconnect the user after the time specified by the RADIUS server.

Idle Timeout Drawn from the RADIUS attribute `Idle-Timeout`, it will disconnect the user after the time specified by the RADIUS server.

Accounting Interval Interim Taken from `Acct-Interim-Interval`, it directs the portal to send interim accounting updates at the specified interval.

URL Redirection Allows the after-authentication redirect URL to be defined by the RADIUS server through `WISPr-Redirection-URL`.

RADIUS Options

These options fine-tune how RADIUS authentication behaves.

NAS Identifier Configures an alternate NAS Identifier to send with RADIUS requests. The default value is `CaptivePortal-<zone name>`.

Reauthentication If enabled, the portal sends `Access-Request` packets to the RADIUS server for each user that is logged in every minute. If an `Access-Reject` is received for a user, that user is disconnected from the captive portal immediately. This allows actively terminating user sessions from the RADIUS server.

> **Warning:** If concurrent login limits are defined in RADIUS this option may not work properly, as the additional request would fail as the reauthentication attempt would be considered a second concurrent login.

> **Note:** If reauthentication is combined with RADIUS accounting, **Interim** accounting updates must be used to track usage during sessions, otherwise the RADIUS server will not know if a user exceeds limits until they logout.

Session-Timeout When set, clients will be disconnected after the amount of time set by the RADIUS `Session-Timeout` attribute sent to the portal at login.

Traffic Quota When set, the portal uses the `pfSense-Max-Total-Octets` reply attribute sent by the RADIUS server to set a traffic quota for a user. This determines an amount of traffic which, when exceeded by a client, will trigger a disconnect of that client by the portal. This includes both upload and download traffic.

Per-user Bandwidth Restrictions When set, the portal uses the `pfSense-Bandwidth-Max-Up` and `pfSense-Bandwidth-Max-Down` reply attribute sent by the RADIUS server to set per-user bandwidth restrictions.

MAC address format This option changes the MAC address format used in RADIUS. Change this to alter the username format for RADIUS MAC authentication to one of the following styles:

Default Colon-separated pairs of digits: `00:11:22:33:44:55`

Single Dash Digits in two groups, separated by a single dash halfway: `001122-334455`

IETF Hyphen-separated pairs of digits: `00-11-22-33-44-55`

Cisco Groups of four digits separated by a period: `0011.2233.4455`

Unformatted All digits together with no formatting or separators: `001122334455`

RADIUS MAC Authentication Options

RADIUS MAC Secret When the portal attempts RADIUS MAC authentication, it sends the MAC Address as the username and this value as the password.

Login Page Fallback When set, the portal will redirect a client to the login page if MAC Authentication failed.

Accounting

RADIUS accounting sends session information back to the RADIUS server indicating when a user session starts, ends, and how much data they have transmitted.

> **Warning:** Not all RADIUS servers support or are configured to accept accounting data. Setup the RADIUS server properly before enabling this feature.

Accounting Server An authentication server entry for a RADIUS server to which the portal will send accounting data (*Authentication Servers*).

Send Accounting Updates Configures the specific type of accounting supported by the server.

No updates Synonymous with disabling accounting, the portal will not send accounting updates to the server.

Stop/start The portal sends START and STOP records for a user session only.

Stop/start (FreeRADIUS) The portal sends START and STOP records for a user session only, in a way that is compatible with FreeRADIUS.

Interim The portal sends START and STOP records and also periodically sends updates to the server while a user session is active. This method is less likely to lose session data if the firewall restarts without notifying the RADIUS server of a STOP message, but will cause increased database usage on the RADIUS server.

Accounting Style When **Invert Acct-Input-Octets and Acct-Output-Octets** is enabled, data counts for RADIUS accounting packets will be taken from the client perspective, not the NAS. `Acct-Input-Octets` will represent download, and `Acct-Output-Octets` will represent upload.

Idle Time Accounting This option changes the time sent in the STOP message for a user disconnected by the portal for idle timeout. When unset (default), the time sent is the last activity time. When set, the idle time is included.

25.3.3 HTTPS login

Login When set, the portal will listen for and accept HTTPS requests for the portal page. This option requires an **SSL/TLS Certificate**.

HTTPS server name The FQDN (hostname + domain) used by the portal for HTTPS. This must match the Common Name (CN) on the certificate to prevent users from receiving certificate errors.

SSL Certificate Select the SSL certificate used by the portal for HTTPS . Certificates are managed in *Certificate Management*.

Disable HTTPS Forwards When checked, attempts by clients to connect to HTTPS sites on port 443 are not redirected to the portal. This prevents users from receiving invalid certificate errors. Users must attempt a connection to an HTTP site, and will then be forwarded to the portal.

25.3.4 HTML Page Contents

When **Use custom captive portal page** is set on the zone, the portal displays these controls to upload custom HTML pages to alter the look of the page presented to users when they are redirected to the portal.

Customizing these pages is optional. Any page contents left blank will use internal defaults.

Portal pages may contain PHP code, and may also include other resources such as images and CSS files. See *File Manager* for more information on including additional assets in a custom portal page.

Warning: Since custom portal pages can run PHP, audit the code to ensure security so the page cannot be exploited by connecting users. Also, avoid granting privileges to this page to untrusted administrators.

In each individual section, the pages can be managed by the displayed controls:

- To upload a new page, click **Browse** and select the file to upload. When the portal options are saved, the file will be copied.

- To view an existing page, click  **View Page Contents**
- To download a copy of an existing page, click  **Download**
- To erase the custom page, click  **Restore Default Page**

Portal page contents

This control is for the main portal page presented to users. Depending on the selected options for the portal, use one of the following examples as the basis for a custom page.

Portal page without authentication

This shows the HTML of a portal page that can be used without authentication.

Listing 1: Download: `example-noauth.html`

```
<html>
<head>
        <title>Welcome to our portal</title>
</head>
<body>
        <p>Welcome to our portal</p>
        <p>Click Continue to access the Internet</p>
        <form method="post" action="$PORTAL_ACTION$">
                <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
                <input name="zone" type="hidden" value="$PORTAL_ZONE$">
                <input name="accept" type="submit" value="Continue">
        </form>
</body>
</html>
```

Portal page with authentication

Here is an example portal page requiring authentication.

Listing 2: Download: `example-auth.html`

```
<html>
<head>
        <title>Welcome to our portal</title>
</head>
<body>
        <p>Welcome to our portal</p>
        <p>Enter your username and password and click Login to access the Internet</p>
        <form method="post" action="$PORTAL_ACTION$">
                <input name="auth_user" type="text">
                <input name="auth_pass" type="password">
                <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
                <input name="zone" type="hidden" value="$PORTAL_ZONE$">
```

(continues on next page)

(continued from previous page)

```
                <input name="accept" type="submit" value="Login">
        </form>
</body>
</html>
```

Portal page with Vouchers

Here is an example portal page for use with vouchers.

Listing 3: Download: `example-voucher.html`

```
<html>
<head>
        <title>Welcome to our portal</title>
</head>
<body>
        <p>Welcome to our portal</p>
        <p>Enter your voucher code and click Login to access the Internet</p>
        <form method="post" action="$PORTAL_ACTION$">
                <input name="auth_voucher" type="text">
                <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
                <input name="zone" type="hidden" value="$PORTAL_ZONE$">
                <input name="accept" type="submit" value="Login">
        </form>
</body>
</html>
```

Authentication error page contents

Using this control, optionally upload a custom HTML page to be displayed when authentication errors happen. An authentication error occurs when a user enters a bad username or password, or in the case of RADIUS authentication, potentially an unreachable RADIUS server.

By default, this error page is simply the login page again.

Logout page contents

The logout page is presented to the user after login and it triggers a popup window. The default code uses JavaScript to create the new window in the following way:

Listing 4: Download: `example-logout.html`

```
<html>
<head><title>Redirecting...</title></head>
<body>
<span style="font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif; font-size:␣
↪11px;">
<b>Redirecting to <a href="<?=$my_redirurl;?>"><?=$my_redirurl;?></a>...</b>
</span>
<script type="text/javascript">
//<![CDATA[
LogoutWin = window.open('', 'Logout', 'toolbar=0,scrollbars=0,location=0,statusbar=0,
↪menubar=0,resizable=0,width=256,height=64');
```

(continues on next page)

(continued from previous page)

```
if (LogoutWin) {
        LogoutWin.document.write('<html>');
        LogoutWin.document.write('<head><title>Logout</title></head>') ;
        LogoutWin.document.write('<body style="background-color:#435370">');
        LogoutWin.document.write('<div class="text-center" style="color: #ffffff;
↪font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif; font-size: 11px;">') ;
        LogoutWin.document.write('<b>Click the button below to disconnect</b><p />');
        LogoutWin.document.write('<form method="POST" action="<?=$logouturl;?>">');
        LogoutWin.document.write('<input name="logout_id" type="hidden" value="<?=
↪$sessionid;?>" />');
        LogoutWin.document.write('<input name="zone" type="hidden" value="<?=$cpzone;?
↪>" />');
        LogoutWin.document.write('<input name="logout" type="submit" value="Logout" />
↪');
        LogoutWin.document.write('</form>');
        LogoutWin.document.write('</div></body>');
        LogoutWin.document.write('</html>');
        LogoutWin.document.close();
}

document.location.href="<?=$my_redirurl;?>";
//]]>
</script>
</body>
</html>
```

Most browsers have pop-up blockers that will most likely stop that logout window from appearing, so investigate other possible means of creating a JavaScript pop-up using similar code.

25.4 MAC Address Control

The **MACs** tab defines actions for MAC addresses that can be either passed through the portal for this zone without requiring authentication, or blocked from reaching the portal.

To manage these MAC entries:

- Navigate to **Services > Captive Portal**
- Click  on the line for the Zone to edit
- Click the **MACs** tab
- Click  **Add** to add a new entry
- Fill in the form as follows:

  Action Defines the action to take on this entry:

  Pass Always allow traffic through from this MAC address without authentication.

  Block Always deny traffic from this MAC address

  MAC address The MAC address of the device to allow. The value must be colon-separated pairs of digits, such as `00:11:22:33:44:55`.

  Description Some text describing the entry, if desired.

Bandwidth up/down The amount of bandwidth that this device may use, specified in Kilobits per second. Leave blank to not specify a limit.

- Click **Save**

From this page, an entry may be edited by clicking  on its row, or deleted by clicking  .

25.5 Allowed IP Address

The **Allowed IP Address** tab works similarly to the **MACs** tab, except it checks IP addresses instead of MAC addresses. Traffic matching the specified IP address and the configured direction will always be allowed through the portal with no authentication in this zone.

IP Address The IP address of the device to always pass through the portal.

Description Some text describing the entry, if desired.

Direction The direction to allow traffic matching this IP address.

From Allow traffic sourced from this IP address through the portal, such as a local client IP address attempting to reach the Internet, or the IP address of a management client that must reach hosts on the portal network.

To Allow traffic with this IP address as a destination, such as a local web server IP address that must be reached via port forward, or a remote web server IP address which clients must always reach.

Both Allow traffic both to and from this IP address.

Bandwidth up/down The amount of bandwidth that this device may use. Leave blank to not specify a limit.

25.6 Allowed Hostnames

Allowed Hostnames work similarly to **Allowed IP Address** entries, except they are configured by hostname instead of IP address. A daemon periodically resolves the hostnames to IP address(es) and allows them through the portal without authentication in this zone.

The most common use of this feature is to make a "walled garden" style portal, where users are permitted to access a restricted set of sites without authenticating to the portal. This is also commonly used with the **Pre-authentication Redirect URL** if that page is hosted externally.

Note: Often sites will use many hostnames, content delivery networks, or ad servers as part of their content. In order to allow a site to load fully, all of these additional sites must be added to the list of allowed hostnames.

Direction The direction to allow traffic matching this hostname. In most typical use cases for allowing hostnames, the *To* or *Both* directions are the best fit.

To Allow traffic from local clients to a remote site matching this hostname as a destination without authentication. For example, a remote web server that must always be reachable by local clients, even when they are not logged in.

From Allow traffic sourced from this hostname through the portal, such as the hostname of a local client attempting to reach the Internet.

Both Allow traffic both to and from this hostname.

Hostname The fully qualified domain name (FQDN) of the target host or site. The hostname must exist in DNS so that it can be resolved to an IP address.

Description Some text describing the entry, if desired.

Bandwidth up/down The amount of bandwidth that traffic to or from this hostname may use. Leave blank to not specify a limit.

25.7 Vouchers

Vouchers are single use codes used to gain Internet access through a Captive Portal. Each roll of vouchers is cryptographically generated and includes a set time limit. Vouchers are common in places where an organization wants authenticated, but time-limited, Internet access without needing to provide a username and password to users.

This type of configuration is prevalent in travel and hospitality locations such as coffee shops, hotels, and airports. Users enter a voucher code in the portal login form and are the portal grants access for the amount of time allowed by the voucher roll. Voucher rolls can be exported by the GUI as a CSV file, and some companies have even integrated the exported voucher lists into point of sale applications to print voucher codes on customer receipts.

Voucher time does not stop counting down if a user logs out; they allow access only from the start of the session until the duration of the voucher length has elapsed. During that time, the voucher can be re-used by the same or a different computer. If the voucher is used again by another computer, the previous session is stopped.

Vouchers require a public/private RSA key pair to generate and verify. A 32-bit set of keys is generated automatically by the firewall the first time the page loads. A custom key pair may be generated manually by the GUI as well. The maximum key length is 64 Bits. Using shorter keys will make the voucher codes shorter but eventually less secure.

25.7.1 Voucher Options

Voucher options are unique per Captive Portal Zone. To configure vouchers, navigate to the **Vouchers** tab when editing a Captive Portal Zone.

Voucher Rolls Voucher rolls are managed by this section of the page. The page lists information about each roll along with a link to add new rolls. No options appear here until after the other settings are active. See *Managing Voucher Rolls* for details.

Enable When checked, this portal zone allows vouchers as a method of authentication, and the page changes to display the other voucher options.

Voucher Keys The keys which generate and verify vouchers. Before vouchers are active on a zone, the GUI randomly generates new values each time the page loads. After saving voucher settings, the keys are present in the configuration and remain static from that point on.

> **Warning:** Do not change the keys or other bits after creating voucher rolls. If the values change, all current voucher rolls are invalid. Create new voucher rolls using the new settings after making changes.

Voucher Public Key This key is used by the portal to decrypt vouchers. Use the existing random key, or click **Generate new keys** to make a new public and private key pair. Users may generate keys elsewhere and paste the RSA public key (64 Bit or smaller) in PEM format here.

Voucher Private Key This key is used by the portal to generate voucher codes and does not need to be available if the vouchers are generated by another system. Use the existing random key or paste in an RSA private key (64 Bit or smaller) in PEM format here.

Character Set The character set defines which characters are valid for voucher text. The character set is case sensitive and should contain printable characters (numbers, lower case and upper case letters) that are hard to confuse with others. For example, avoid `0` (Digit zero), `O` (Letter O), `l` (Lowercase L), and `1` (Digit One). It cannot contain a space, double quote, or comma. A smaller character set will result in longer vouchers to ensure sufficient randomness.

Voucher Bits The following "bit" fields control how the vouchers themselves are generated by the portal. The best practice is to leave these values at their defaults, but they may be adjusted if necessary. The total of all bit fields *must* be less than the RSA key size. For example, the default values are `16`, `10`, and `5`. The sum of these is `31`, which is one less than the default RSA key size of `32`.

# of Roll Bits Number of bits for the Roll ID. Set this larger to have a lot of rolls active at the same time. Can be from `1-31`, the default value is `16`.

# of Ticket Bits Number of bits for the Ticket ID. Set this larger if each roll will have a large number of vouchers. Can be from `1-16`, the default value is `10`.

# of Checksum Bits Reserves a range in each voucher to store a simple checksum over **Roll bits** and **Ticket bits**. Allowed range is `0-31`, the default value is `5`.

Magic Number The magic number is present in every voucher, and is verified by the portal during voucher check. The size of the magic number depends on the number of bits remaining after adding together the number of bits for the roll, ticket, and checksum. If there are no bits remaining for use by the magic number, then the portal does not use magic numbers.

Invalid Voucher Message This message is displayed by the portal to the user if they attempt to enter a voucher that does not exist or is not valid in any way except for being expired.

Expired Voucher Message This message is displayed to the user by the portal if they enter a voucher that was valid, but has expired.

25.7.2 Enable Vouchers

- Use the pfSense® WebGUI to navigate to **Services > Captive Portal**
- Click  on the line for the Zone to edit
- Ensure the Zone **Authentication Method** is set to *Use an Authentication backend*, change the value and save if necessary.
- Click the **Vouchers** tab
- Check **Enable**
- Fill in the form based on the options described in *Voucher Options*. In most cases, the options may remain at their default values.
- Click **Save**

With Vouchers enabled, the voucher management controls will be active in the GUI.

25.7.3 Managing Voucher Rolls

Vouchers are created by the portal in batches called **Rolls**. Each roll has specific settings that are unique to that roll. For example, a roll can have an 8-hour time limit and a separate roll can have a 12-hour time limit. Then users may be given voucher codes depending on which level of service they purchased and they will be limited to the amount of time corresponding to the voucher roll from which their code was picked.

Voucher Roll Options

Roll # The number of this roll. Each roll must have a unique number. This can be any number from `0` to `65535` with the default number of **Roll Bits**.

Minutes per Ticket Defines how long the voucher lasts, in minutes. The voucher time starts counting down the moment the voucher is used, and does not stop, so plan the voucher length accordingly. Because this is defined in minutes, ensure the correct length is used, e.g. `1440` minutes is 24 hours.

Count Defines the number of vouchers in this roll. The value can be from `0` to `1023` with the default number of **Ticket Bits**.

Note: If the count on an existing roll is changed, it will invalidate **all** other vouchers on the roll.

Comment A description of the roll for reference, such as `2 hour vouchers for coffee purchases`.

Creating Voucher Rolls

To create a voucher roll:

- Use the pfSense® WebGUI to navigate to **Services > Captive Portal**
- Click  on the line for the Zone to edit
- Click the **Vouchers** tab
- Click  **Add** under the roll list
- Fill in the options as described in *Voucher Roll Options*
- Click **Save**

The new roll is available for immediate use by clients.

Editing Existing Rolls

To edit an existing voucher roll, click  at the end of its row, but be careful when making changes. Changing the **Roll** number or **Count** will invalidate the current vouchers on the roll.

Removing Voucher Rolls

To remove rolls of vouchers, click  at the end of their row. When a roll is removed, *all* of the vouchers in that roll become invalid. Do not remove a roll unless it has been completely used, compromised, or otherwise unnecessary.

Exporting/Downloading Voucher Rolls

Click  to download a file containing the vouchers in the specified roll. This action downloads a `.csv` (Comma Separated Value) spreadsheet containing all voucher codes for this roll.

Nearly any spreadsheet editor can open this file, such as LibreOffice Calc, Google Docs, or Excel. Programs such as those can print vouchers, feed them into a POS system, and so on.

Using Vouchers on A Portal Page

The portal page must submit voucher codes via the `auth_voucher` form field. See *Portal page with Vouchers* for an example.

Viewing Active Vouchers

To view the list of currently active vouchers and their timers, navigate to **Status > Captive Portal**, on the **Active Vouchers** tab for a zone, as seen in Figure *Active Vouchers*.

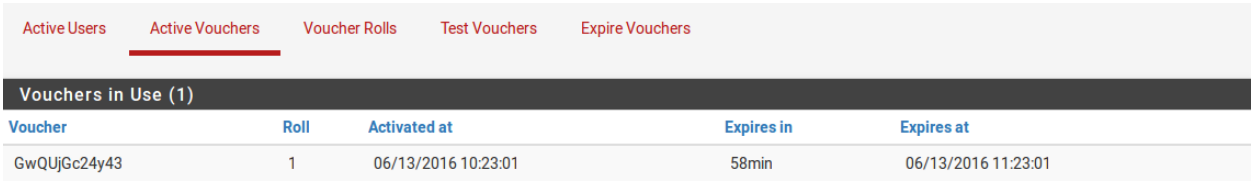


Fig. 1: Active Vouchers

Viewing Voucher Roll Utilization

To view a list of voucher rolls and usage counts, navigate to **Status > Captive Portal**, on the **Voucher Rolls** tab for a zone, as in Figure *Vouchers Roll Usage*

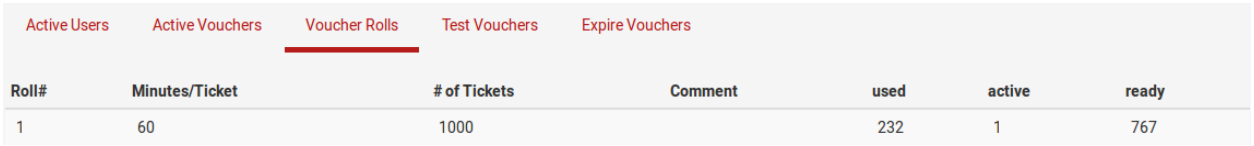


Fig. 2: Vouchers Roll Usage

Testing Vouchers

To test the validity of a voucher code, enter it at **Status > Captive Portal**, on the **Test Vouchers** tab for a zone. Upon submission, the page will display if a code is valid or not, and if it is valid, it will show the voucher time limit, as seen in Figure *Testing Vouchers*. Testing a voucher does not count it as used or expired, it is still free to be used at a later time by a client.

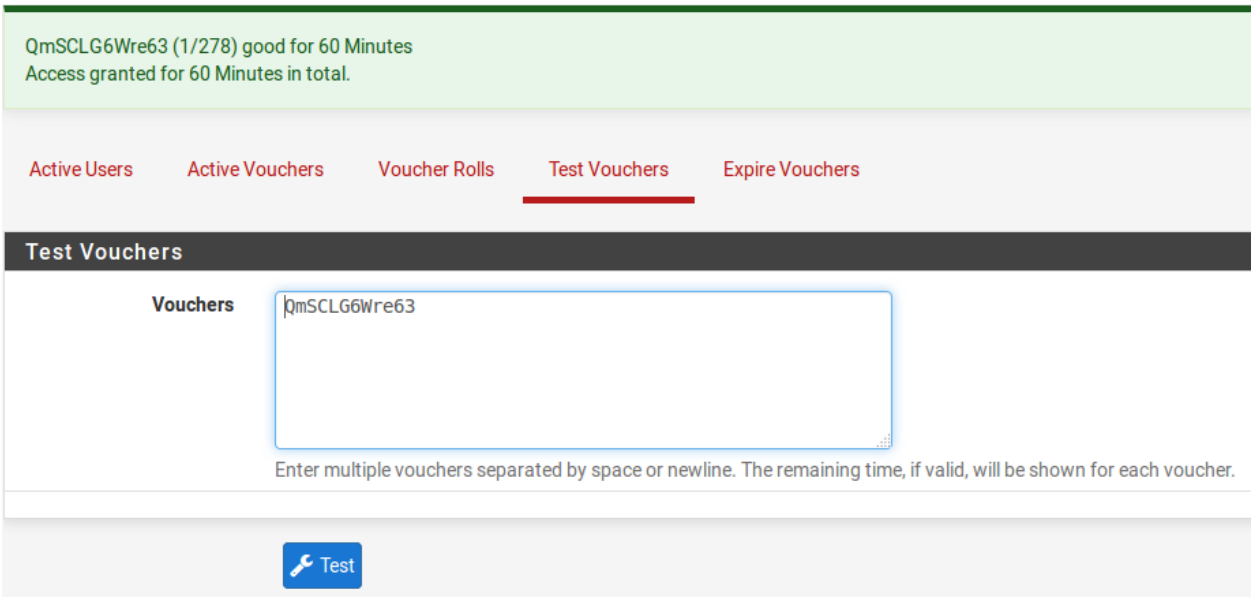


Fig. 3: Testing Vouchers

Expiring Vouchers

To invalidate vouchers, during or before use, enter them at **Status > Captive Portal**, on the **Expire Vouchers** tab for a zone. After submitting, any voucher listed in the form will no longer be allowed by the portal. Active vouchers entered on this page are also immediately expired.

The page can expire any number of vouchers in a batch. Enter vouchers separated by newlines.

25.7.4 Synchronizing Vouchers

At the bottom of the **Vouchers** tab there are options to synchronize vouchers to another HA node. This works similarly to the XML-RPC configuration synchronization found in high availability setups (See *pfSense XML-RPC Config Sync Overview*). When active, this function copies the voucher rolls to the target node and also pushes information about active vouchers to the target node as vouchers are consumed by users.

Synchronize Voucher Database IP The target IP address or hostname of the other node for voucher synchronization.

Voucher sync port The port on the target node where the GUI is listening (Typically `443`).

Voucher sync username The username for synchronization access (Must have appropriate privileges).

Voucher sync password The GUI password for the target node.

Unlike the configuration synchronization in a high availability cluster, this synchronization is configured on both the primary and secondary nodes. This ensures that vouchers used on the secondary node while it is active are also sent

back to the primary when it returns to an active state. Unlike configuration synchronization, this does not create a loop.

25.8 File Manager

The **File Manager** tab in a Captive Portal zone is used to upload files that can then be utilized inside a captive portal page, such as style sheets, image files, PHP or JavaScript files.

The total size limit for all files in a zone is **1 MB**.

25.8.1 File Name Conventions

When a file is uploaded using the File Manager, the file name will automatically be prefixed with `captiveportal-`. For example, if `logo.png` is uploaded it will become `captiveportal-logo.png`. If a file already has that prefix in its name, the name is not changed.

These files will be made available in the root directory of the captive portal server for this zone. The files may be referenced directly from the portal page HTML code using relative paths.

Example: An image with the name `captiveportal-logo.jpg` was uploaded using the file manager, It can then be included in the portal page as follows:

```
<img src="captiveportal-logo.jpg" />
```

PHP scripts may be uploaded as well, but they may need to be passed extra parameters to work as desired, for example:

```
<a href="/captiveportal-aup.php?zone=$PORTAL_ZONE$&redirurl=$PORTAL_REDIRURL$">
    Acceptable usage policy
</a>
```

25.8.2 Managing Files

To upload files:

- Navigate to **Services > Captive Portal**
- Edit the zone where the files will be uploaded
- Click the **File Manager** tab
- Click 
- Click **Browse**
- Locate and select the file to upload
- Click  **Upload**

The file will be transferred to the firewall and stored in the configuration.

To delete files:

- Navigate to **Services > Captive Portal**
- Edit the zone where the file to delete is located

- Click the **File Manager** tab
- Click  next to the file to remove
- Click **OK** to confirm the delete action

The file will be removed from the portal configuration and will no longer be available for use in portal pages.

See also:

- *Captive Portal Status*
- *Captive Portal Authentication Logs*
- *Troubleshooting Captive Portal*

Captive Portal in pfSense® software forces users on an interface to authenticate before granting access to the Internet. Where possible, the firewall automatically presents a login web page in which the user must enter credentials such as a username/password, a voucher code, or a simple click-through agreement.

This feature is commonly used throughout the hospitality industry (Hotels, restaurants, airports, and more) as well as in corporate and even home environments. It is primarily used for wireless hot spots or for additional authentication before allowing access to internal networks from wireless clients.

Captive Portal is configured under **Services > Captive Portal**.

See also:

pfSense Hangouts on Youtube to view the April 2015 Hangout on Captive Portal

25.9 Limitations

The Captive Portal implementation in pfSense does have some limitations. This section covers those, and the common ways of working around them where possible.

25.9.1 Does not yet support IPv6

Currently, Captive Portal does not support IPv6.

25.9.2 Not capable of reverse portal

A reverse portal, requiring authentication for traffic coming into a local network from the Internet, is not possible.

CHAPTER

TWENTYSIX

HIGH AVAILABILITY

26.1 pfsync Overview

pfsync enables the synchronization of the firewall state table between cluster nodes. Changes to the state table on the primary are sent to the secondary firewall(s) over the Sync interface, and vice versa. When pfsync is active and properly configured, all nodes will have knowledge of each connection flowing through the cluster. If the master node fails, the backup node will take over and clients will not notice the transition since both nodes knew about the connection beforehand.

pfsync uses multicast by default, though an IP address can be defined to force unicast updates for environments with only two firewalls where multicast traffic will not function properly. Any active interface can be used for sending pfsync updates, however utilizing a dedicated interface is better for security and performance. pfsync does not support any method of authentication, so if anything other than a dedicated interface is used, it is possible for any user with local network access to insert states into the state table. In low throughput environments that aren't security paranoid, use of the LAN interface for this purpose is acceptable. Bandwidth required for this state synchronization will vary significantly from one environment to another, but could be as high as 10% of the throughput traversing the firewall depending on the rate of state insertions and deletions in a network.

Failover can still operate without pfsync, but it will not be seamless. Without pfsync if a node fails and another takes over, user connections would be dropped. Users may immediately reconnect through the other node, but they would be disrupted during the transition. Depending on the usage in a particular environment, this may go unnoticed or it could be a significant, but brief, outage.

When pfsync is in use, pfsync settings *must be enabled* on all nodes participating in state synchronization, including secondary nodes, or it will not function properly.

26.1.1 pfsync and Firewall Rules

Traffic for pfsync must be explicitly passed on the Sync interface. The rule must pass the *pfsync* protocol from a source of the *Sync network* to *any* destination. A rule passing all traffic of any protocol would also allow the required traffic, but a more specific rule is more secure.

26.1.2 pfsync and Physical Interfaces

States in pfSense® are bound to specific operating system Interfaces. For example, if WAN is *em0*, then a state on WAN would be tied to *em0*. If the cluster nodes have identical hardware and interface assignments then this works as expected. In cases when different hardware is used, this can be a problem. If WAN on one node is *em0* but on another node it is *igb0*, the states will not match and they will not be treated the same.

It is always preferable to have identical hardware, but in cases where this is impractical there is a workaround: Adding interfaces to a LAGG will abstract the actual underlying physical interface so in the above example, WAN would be *lagg0* on both and states would be bound to *lagg0*, even though *lagg0* on one node contains *em0* and it contains *igb0* on the other node.

26.1.3 pfsync and Upgrades

Normally pfSense would allow firewall upgrades without any network disruption. Unfortunately, this isn't always the case with upgrades as the pfsync protocol can change to accommodate additional functionality. Always check the upgrade guide linked in all release announcements before upgrading to see if there are any special considerations for CARP users.

26.2 pfSense XML-RPC Config Sync Overview

To make the job of maintaining practically identical pfSense® software nodes easier, configuration synchronization is possible using XML-RPC. When XML-RPC Synchronization is enabled, settings from supported areas are copied to the secondary and activated after each configuration change. XMLRPC Synchronization is optional, but maintaining a cluster is a lot more work without it.

Some areas cannot be synchronized, such as the Interface configuration, but many other areas can: Firewall rules, aliases, users, certificates, VPNs, DHCP, routes, gateways, and more. As a general rule, items specific to hardware or a particular installation, such as Interfaces or values under **System > General** or **System > Advanced** do not synchronize. The list of supported areas can vary depending on the version of pfSense in use. For a list of areas that will synchronize, see the checkbox items on **System > High Avail Sync** in the XMLRPC section. Most packages will not synchronize but some contain their own synchronization settings. Consult package documentation for more details.

Configuration synchronization should use the Sync interface, or if there is no dedicated Sync interface, use the same interface configured for pfsync.

In a two-node cluster the XML-RPC settings must *only* be enabled on the primary node, the secondary node must have these settings *disabled*.

For XML-RPC to function, both nodes must meet the following requirements:

- The GUI must be running on the same port and protocol, for example: HTTPS on port `443`, which is the default setting.
- The interfaces must be assigned identically on both nodes, for example: wan=WAN, lan=LAN, opt1=Sync, opt2=DMZ. Check the `config.xml` contents directly to ensure a match.

> **Warning:** If the interfaces do not match up exactly, firewall rules and other configuration items will appear to synchronize to the wrong interface on the secondary node. Additionally, this can also lead to failures in DHCP failover.

- The sync user must either be `admin` or an account with the *System - HA node sync* privilege.

> **Note:** If XML-RPC will synchronize users, create the sync user on the secondary manually first, as well as on the primary. The redundant copy on the secondary will be removed during the first successful synchronization, but the initial synchronization cannot succeed without it.

26.3 Verifying Failover Functionality

Since using HA is about high availability, thorough testing before placing a cluster into production is a must. The most important part of that testing is making sure that the HA peers will failover gracefully during system outages.

If any actions in this section do not work as expected, see *Troubleshooting High Availability*.

26.3.1 Check CARP status

On both systems, navigate to **Status > CARP (failover)**. If everything is working correctly, the primary will show **MASTER** for the status of all CARP VIPs and the secondary will show **BACKUP**.

If either instead shows **DISABLED**, click the **Enable CARP** button and then refresh the page.

If an interface shows **INIT**, it means the interface containing the CARP VIP does not have a link. Connect the interface to a switch, or at least to the other node. If the interface will not be used for some time, remove the CARP VIP from the interface as this will interfere with normal CARP operation.

26.3.2 Check Configuration Replication

Navigate to key locations on the secondary node, such as **Firewall > Rules** and **Firewall > NAT** and ensure that rules created only on the primary node are being replicated to the secondary node.

If the example earlier in this chapter was followed, the "temp" firewall rule on the pfsync interface would be replaced by the rule from the primary.

26.3.3 Check DHCP Failover Status

If DHCP failover was configured, its status can be checked at **Status > DHCP Leases**. A new section will appear at the top of the page containing the status of the DHCP Failover pool, as in Figure *DHCP Failover Pool Status*.

Pool Status

Failover Group	My State	Since	Peer State	Since
dhcp_lan (LAN)	normal	2016/03/15 18:52:51	normal	2016/03/15 18:52:51

Fig. 1: DHCP Failover Pool Status

26.3.4 Test CARP Failover

Now for the real failover test. Before starting, make sure that a local client behind the CARP pair on LAN can connect to the Internet with both pfSense® firewalls online and running. Once that is confirmed to work, it is an excellent time to make a backup.

For the actual test, unplug the primary node from the network or shut it down temporarily. The client will be able to keep loading content from the Internet through the secondary node. Check **Status > CARP (failover)** again on the backup and it will now report that it is **MASTER** for the LAN and WAN CARP VIPs.

Now bring the primary node back online and it will regain its role as **MASTER**, and the backup system will demote itself to **BACKUP** once again. At any point during this process, Internet connectivity will still work properly.

Test the HA pair in as many failure scenarios as possible. Additional tests include:

- Unplug the WAN or LAN cable
- Pull the power plug of the primary
- Disable CARP on the primary using both the temporary disable feature and maintenance mode
- Test with each system individually (power off secondary, then power back on and shut down the primary)
- Download a file or try streaming audio/video during the failover
- Run a continuous ICMP echo request (ping) to an Internet host during the failover

26.4 Layer 2 Redundancy

The diagrams earlier in this chapter did not describe layer 2 (switch) redundancy, to avoid throwing too many concepts at readers simultaneously. This section covers the layer 2 design elements to be considered when planning a redundant network. This chapter assumes a two system deployment, though this scales to as many installations as required.

If both redundant pfSense® firewalls are plugged into the same switch on any interface, that switch becomes a single point of failure. To avoid this single point of failure, the best choice is to deploy two switches for each interface (other than the dedicated pfsync interface).

Example HA Network Diagram is network-centric, not showing the switch infrastructure. The Figure *Diagram of HA with Redundant Switches* illustrates how that environment looks with a redundant switch infrastructure.

26.4.1 Switch Configuration

When using multiple switches, the switches should be interconnected. As long as there is a single connection between the two switches, and no bridge on either of the firewalls, this is safe with any type of switch. Where using bridging, or where multiple interconnections exist between the switches, care must be taken to avoid layer 2 loops. A managed switch would be required which is capable of using Spanning Tree Protocol (STP) to detect and block ports that would otherwise create switch loops. When using STP, if an active link dies, e.g. switch failure, then a backup link can automatically be brought up in its place.

pfSense also has support for *lagg(4)* link aggregation and link failover interfaces which allows multiple network interfaces to be plugged into one or more switches for increased fault tolerance. See *LAGG (Link Aggregation)* for more information on configuring link aggregation.

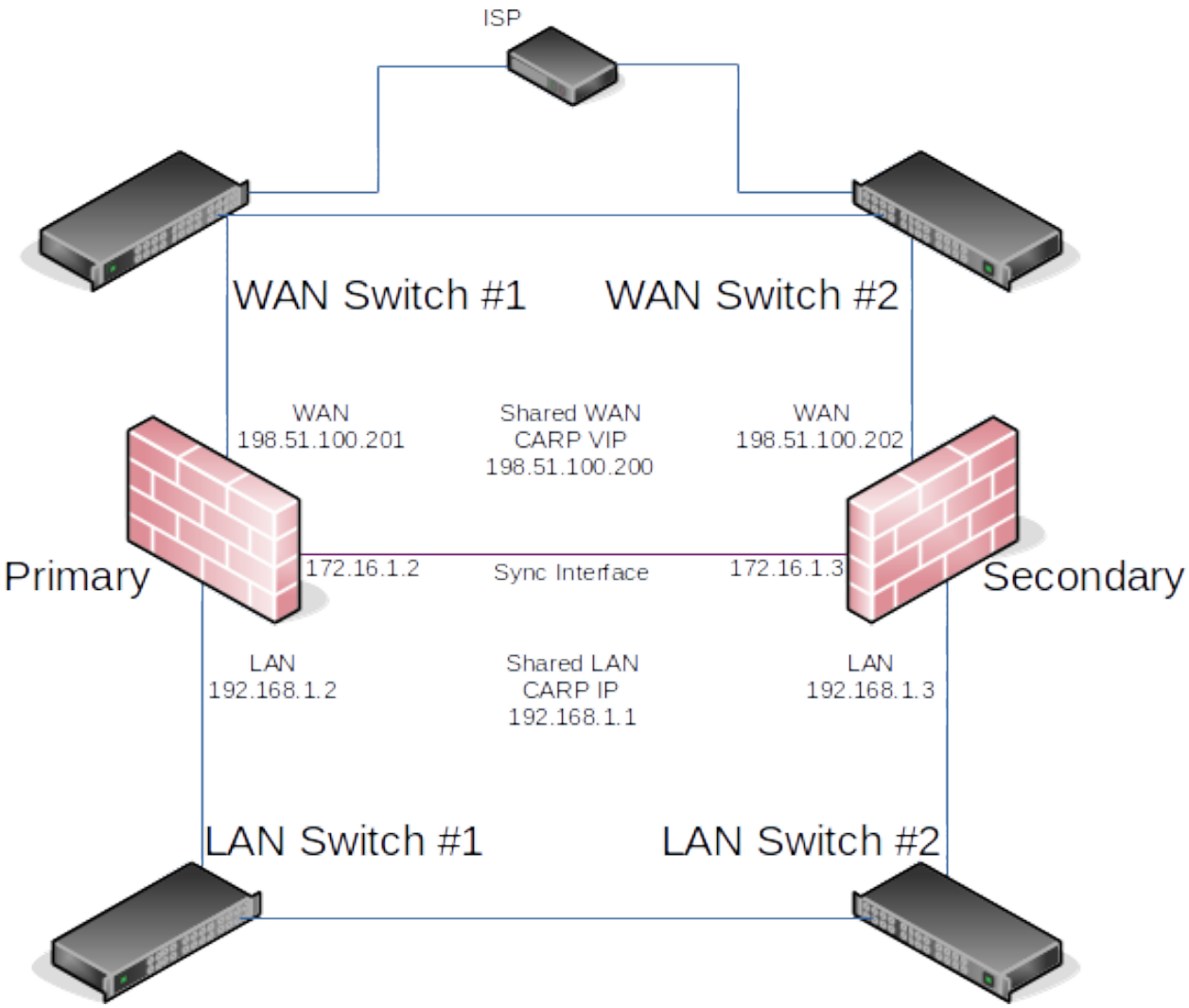


Fig. 2: Diagram of HA with Redundant Switches

26.4.2 Host Redundancy

It is more difficult to obtain host redundancy for critical systems inside the firewall. Each system could have two network cards and a connection to each group of switches using Link Aggregation Control Protocol (LACP) or similar vendor-specific functionality. Servers could also have multiple network connections, and depending on the OS it may be possible to run CARP or a similar protocol on a set of servers so that they would be redundant as well. Providing host redundancy is more specific to the capabilities of the switches and server operating systems, which is outside the scope of this documentation.

26.4.3 Other Single Points of Failure

When trying to design a fully redundant network, there are many single points of failure that sometimes get missed. Depending on the level of uptime to achieve, there are more and more things to consider than a simple switch failure. Here are a few more examples for redundancy on a wider scale:

- Supply isolated power for each redundant segment.
  - Use separate breakers for redundant systems.
  - Use multiple UPS banks/generators.
  - Use multiple power providers, entering opposite sides of the building where possible.
- Even a Multi-WAN configuration is no guarantee of Internet uptime.
  - Use multiple Internet connection technologies (DSL, Cable, Fiber, Wireless).
  - If any two carriers use the same pole/tunnel/path, they could both be knocked out at the same time.
- Have backup cooling, redundant chillers or a portable/emergency air conditioner.
- Consider placing the second set of redundant equipment in another room, another floor, or another building.
- Have a duplicate setup in another part of town or another city.
- I hear hosting is cheap on Mars, but the latency is killer.

26.5 High Availability with Bridging

High availability is not currently compatible with bridging in a native capacity that is considered reliable or worthy of production use. It requires significant manual intervention. The details of the process can be found in *High Availability*.

26.6 Using IP Aliases to Reduce Heartbeat Traffic

If there are a large number of CARP VIPs on a segment, this can lead to a lot of multicast traffic. One heartbeat per second is sent per CARP VIP. To reduce this traffic, additional VIPs may be "stacked" on top of one CARP VIP on an interface. First, pick one CARP VIP to be the "main" VIP for the interface. Then, change the other CARP VIPs in that same subnet to be an *IP Alias* type VIP, with the "main" CARP VIP interface selected to be their **Interface** on the VIP configuration.

This not only reduces the heartbeats that will be seen on a given segment, but it also causes all of the IP alias VIPs to change status along with the "main" CARP VIP, reducing the likelihood that a layer 2 issue will cause individual CARP VIPs to not fail over as expected.

IP Alias VIPs do not normally synchronize via XML-RPC configuration synchronization, however, IP alias VIPs set to use CARP interfaces in this manner will synchronize.

See also:

- *Upgrading High Availability Clusters*
- *High Availability Configuration Example*
- *High Availability Configuration Example with Multi-WAN*
- *High Availability Configuration Example without NAT*
- *Troubleshooting High Availability*
- *Troubleshooting High Availability DHCP Failover*
- *Troubleshooting VPN Connectivity to a High Availability Secondary Node*

pfSense® software is one of very few open source solutions offering enterprise-class high availability capabilities with stateful failover, allowing the elimination of the firewall as a single point of failure. High Availability is achieved through a combination of features:

- CARP for IP address redundancy
- XMLRPC for configuration synchronization
- pfsync for state table synchronization

With this configuration, nodes act as an "active/passive" cluster with the primary node working as the master node and the secondary node in a backup role, taking over as needed if the primary node fails.

Though often erroneously called a "CARP Cluster", two or more redundant firewalls are more aptly titled a "High Availability Cluster" or "HA Cluster", since CARP is only one of several technologies used to achieve High Availability with pfSense, and in the future CARP could be swapped for a different redundancy protocol.

One interface on each cluster node will be dedicated for synchronization tasks. This is typically referred to as the "Sync" interface, and it is used for configuration synchronization and pfsync state synchronization. Any available interface may be used.

Note: Some call this the "CARP" interface but that is incorrect and very misleading. CARP heartbeats happen on each interface with a CARP VIP; CARP traffic and failover actions do not utilize the Sync interface.

The most common High Availability cluster configuration includes only two nodes. It is possible to have more nodes in a cluster, but they do not provide a significant advantage.

It is important to distinguish between the three functions (IP address redundancy, configuration synchronization, and state table synchronization), because they happen in different places. Configuration synchronization and state synchronization happen on the sync interface, directly communicating between firewall units. CARP heartbeats are sent on each interface with a CARP VIP. Failover signaling does not happen on the sync interface, but rather it happens on every CARP-enabled interface.

See also:

pfSense Hangouts on Youtube which contains the June 2015 Hangout also covering High Availability.

26.7 CARP Overview

Common Address Redundancy Protocol (CARP) was created by OpenBSD developers as a free, open redundancy solution for sharing IP addresses among a group of network devices. Similar solutions already existed, primarily the IETF standard for Virtual Router Redundancy Protocol (VRRP). However Cisco claims VRRP is covered by its patent on their Hot Standby Router Protocol (HSRP), and told the OpenBSD developers that it would enforce its patent. Hence, the OpenBSD developers created a new free, open protocol to accomplish essentially the same result without infringing on Cisco's patent. CARP became available in October 2003 in OpenBSD, and was later added to FreeBSD as well.

A CARP type Virtual IP address (VIP) is shared between nodes of a cluster. One node is master and receives traffic for the IP address, and the other nodes maintain backup status and monitor for heartbeats to see if they need to assume the master role if the previous master fails. Since only one member of the cluster at a time is using the IP address, there is no IP address conflict for CARP VIPs.

In order for failover to work properly it is important that inbound traffic coming to the cluster, such as routed upstream traffic, VPNs, NAT, local client gateway, DNS requests, etc., be sent to a CARP VIP and for outgoing traffic such as Outbound NAT to be sent from a CARP VIP. If traffic is addressed to a node directly and not a CARP VIP, then that traffic will not be picked up by other nodes.

CARP works similar to VRRP and HSRP, and may even conflict in some cases. Heartbeats are sent out on each interface containing a CARP VIP, one heartbeat per VIP per interface. At the default values for skew and base, a VIP sends out heartbeats about once per second. The skew determines which node is master at a given point in time. Whichever node transmits heartbeats the fastest assumes the master role. A higher skew value causes heartbeats to be transmitted with more delay, so a node with a lower skew will be the master unless a network or other issue causes the heartbeats to be delayed or lost.

Note: Never access the firewall GUI, SSH, or other management mechanism using a CARP VIP. For management purposes, only use the actual IP address on the interface of each separate node and not the VIP. Otherwise it cannot be determined beforehand which unit is being accessed.

26.7.1 IP Address Requirements for CARP

A High Availability cluster using CARP needs three IP addresses in each subnet along with a separate unused subnet for the Sync interface. For WANs, this means that a /29 subnet or larger is required for an optimal configuration. One IP address is used by each node, plus a shared CARP VIP address for failover. The synchronization interface only requires one IP address per node.

It is technically possible to configure an interface with a CARP VIP as the only IP address in a given subnet, but it is not generally recommended. When used on a WAN, this type of configuration will only allow communication from the primary node to the WAN, which greatly complicates tasks such as updates, package installations, gateway monitoring, or anything that requires external connectivity from the secondary node. It can be a better fit for an internal interface, however internal interfaces do not typically suffer from the same IP address limitations as a WAN, so it is still preferable to configure IP addresses on all nodes.

26.7.2 Switch/Layer 2 Concerns

CARP heartbeats utilize multicast and may require special handling on the switches involved with the cluster. Some switches filter, rate limit, or otherwise interfere with multicast in ways that can cause CARP to fail. Also, some switches employ port security methods which may not work properly with CARP.

At a minimum, the switch must:

- Allow Multicast traffic to be sent and received without interference on ports using CARP VIPs.
- Allow traffic to be sent and received using multiple MAC addresses.
- Allow the CARP VIP MAC address to move between ports.

Nearly all problems with CARP failing to properly reflect the expected status are failures of the switch or other layer 2 issues, so be sure the switches are properly configured before continuing.

CHAPTER

TWENTYSEVEN

SYSTEM MONITORING

The data and information that pfSense® software collects and displays is every bit as important as the services it provides. Sometimes it seems that commercial routers go out of their way to hide as much information as possible from users, but pfSense can provide almost as much information as anyone could ever want (and then some).

This chapter contains a variety of methods for finding information about the firewall status, logs, traffic, hardware, and so on.

27.1 Status

These articles cover various ways to check the status of services or features of the firewall, or the firewall itself.

27.1.1 Dashboard

The main page of the firewall is the **Dashboard**. The Dashboard page provides a wealth of information that can be seen at a glance, contained in configurable widgets. These widgets can be added or removed, and dragged around into different positions.

Managing Widgets

Each widget follows some basic conventions for controlling its position, size, settings, and so on, the mechanics of which are covered here in this section.

Adding and Removing Widgets

To start adding widgets, click the  button in the Dashboard controls area of the breadcrumb bar to display the list of available widgets. See *Dashboard Controls in the Breadcrumb Bar*.



Fig. 1: Dashboard Controls in the Breadcrumb Bar

Inside the **Available Widgets** panel, click on the name of a widget to add it to the Dashboard (See *Available Widgets List*). The dashboard will reload with the new widget displayed in one of its columns.

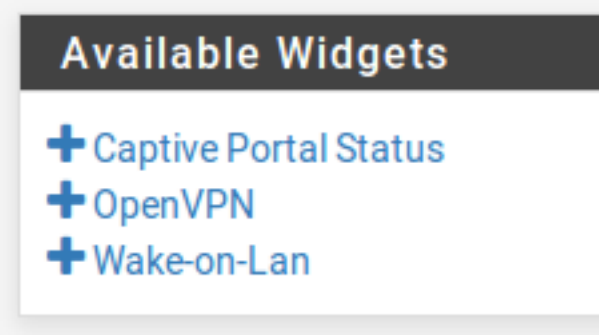


Fig. 2: Available Widgets List

To close and remove a widget from the Dashboard, click the button in its title bar, as seen in Figure *Widget Title Bar*, then click in the dashboard controls.



Fig. 3: Widget Title Bar

Rearranging Widgets

Widgets can be rearranged and moved between columns. To move a widget, click and drag its title bar (Figure *Widget Title Bar*), move the mouse to the desired position, and then release. As the widget is moved it will "snap" into its new position, so the new location may be previewed before releasing the mouse button. After positioning a widget, click in the dashboard controls (*Dashboard Controls in the Breadcrumb Bar*).

Minimizing Widgets

To minimize a widget so it hides its content and only shows up as its title bar, click the button in its title bar, as seen in Figure *Widget Title Bar*. To restore the widget to its normal display, click the button. After changing the widget status, click in the dashboard controls (*Dashboard Controls in the Breadcrumb Bar*).

Changing Widget Settings

Some widgets have customizable settings that control how their content is displayed or updated. If a widget has settings, the  button will show up in its title bar as seen in Figure *Widget Title Bar*. Click that button and the settings for the widget will appear. Once the settings have been adjusted, click the **Save** button inside of the widget settings panel.

Available Widgets

Each widget contains a specific set of data, type of information, graph, etc. Each of the currently available widgets will be covered in this section, along with their settings (if any). These are listed in alphabetical order.

Captive Portal Status

This widget shows the current list of online captive portal users, including their IP address, MAC address, and username.

CARP Status

The CARP Status widget displays a list of all CARP type Virtual IP addresses, along with their status as either MASTER or BACKUP.

Dynamic DNS

The Dynamic DNS widget displays a list of all configured Dynamic DNS hostnames, their current address, and status.

Gateways

The Gateways widget lists all of the system gateways along with their current status. The status information consists of the gateway IP address, Round Trip Time (RTT) also known as delay or latency, the amount of packet loss, and the status (Online, Warning, Down, or Gathering Data). The widgets is updated every few seconds via AJAX.

Gmirror Status

This widget will show the status of a gmirror RAID array on the system, if one is configured. The widget will show if the array is online/OK (Complete), rebuilding, or degraded.

Installed Packages

The Installed Packages widget lists all of the packages installed on the system, along with some basic information about them such as the installed version and whether or not an update is available.

When a package has an update available,  is displayed next to the version number. Packages may be updated from this widget by clicking the  button at the end of a package's row.

Packages may also be reinstalled by clicking  or removed by clicking  .

Interface Statistics

This widget shows a grid, with each interface on the system shown in its own column. Various interface statistics are shown in each row, including packet, byte, and error counts.

Interfaces

The **Interfaces** widget differs from the **Interface Statistics** widget in that it displays general information about the interface rather than counters. The **Interfaces** widget shows the type and name of each interface, IPv4 address, IPv6 address, the interface link status (up or down), as well as the link speed when available.

IPsec

The IPsec widget has three tabs: The first tab, **Overview**, is a count of active and inactive tunnels. The second tab, **Tunnel Status**, lists each configured IPsec tunnel and whether that tunnel is up or down. The last tab, **Mobile**, shows online remote access IPsec VPN users, such as those using IKEv2 or Xauth.

Load Balancer Status

This widget displays a compact view of the server Load Balancing setup. Each row shows the status for one virtual server. The **Server** column shows the virtual server name, status, and IP address with port where the virtual server is accepting connections. The **Pool** column shows the individual pool servers and their status, with an uptime percentage. The **Description** column shows the text description from the virtual server.

Firewall Logs

The **Firewall Logs** widget provides an AJAX-updating view of the firewall log. The number of rows shown by the widget is configurable. As with the normal firewall log view, clicking the action icon next to the log entry will show a window displaying which rule caused the log entry. Clicking the source or destination IP address will copy that value to **Diagnostics > DNS** where the address can be resolved.

NTP Status

The **NTP Status** widget shows the current NTP synchronization source and the server time from that source.

OpenVPN

The **OpenVPN** widget displays the status of each configured OpenVPN instance, for both servers and clients. The status of each instance is shown, but the style and type of information shown varies depending on the type of OpenVPN connection. For example, SSL/TLS based servers show a list of all connected clients. For static key clients and servers, an up/down status is displayed. In each case it displays the IP address of the connecting client with the name and time of the connection.

Picture

The **Picture** widget, as the name implies, displays a picture chosen by the user. This can either be used functionally, for a network diagram or similar, or it can be for style, displaying a company logo or other image.

To add an image:

- Click  on the Picture widget title bar
- Click **Browse** to locate the picture to upload
- Click **Upload** to upload the picture

The size of the picture will adjust to fit the area of the widget, which can vary depending on the size of the browser and platform.

RSS

The RSS (RDFSite Summary, or as it's often called, Really Simple Syndication) widget will display an arbitrary RSS feed. By default, it shows the pfSense® blog RSS feed. Some people choose to show internal company RSS feeds or security site RSS feeds, but it can load any RSS feed.

In addition to defining the RSS feeds to display, the number of stories and size of displayed content are also configurable.

Services Status

This widget provides the same view and control of services that appears under **Status > Services**. Each service is listed along with its description, status (Running, Stopped), and start/restart/stop controls.

SMART Status

If S.M.A.R.T. is enabled on a drive in the firewall, this widget will show a brief status of the drive integrity as reported by S.M.A.R.T.

System Information

This widget is the main widget, displaying a wide array of information about the running system. The information displayed includes:

Name The configured fully qualified hostname of the firewall.

Version The current running version of pfSense on the firewall. The version, architecture, and build time are displayed at the top. Under the build time, the underlying version of FreeBSD is shown.

Under those items is the result of an automatic update check for a more recent version of pfSense software. This automatic update check can be disabled in the update settings.

CPU Type The displayed CPU type is the version string for the processor, such as "Intel(R) Atom(TM) CPU C2758 @ 2.40GHz". The CPU count and package/core layout is also displayed.

If **powerd** is active and the CPU frequency has been lowered, then the current frequency is shown along size the maximum frequency.

Hardware crypto If a known hardware cryptographic accelerator has been detected, it will be displayed here.

Uptime This is the time since the firewall was last rebooted.

Current date/time The current date and time of the firewall, including the time zone. This is useful for comparing the log entries, especially when the time zone on the firewall is different from where the user resides.

DNS Server(s) Lists all of the configured DNS Servers on the firewall.

Last config change The date of the last configuration change on the firewall.

State table size Shows a graphical and numerical representation of active states and the maximum possible states as configured on the firewall. Underneath the state counts is a link to view the contents of the state table.

MBUF usage Shows the number of network memory buffer clusters in use, and the maximum the system has available. These network memory buffers are used for network operations, among other tasks. If the number is close to maximum or at the maximum, increase the number of available mbufs as described in *Hardware Tuning and Troubleshooting*.

Load Average A count of how many active processes are running on the firewall during the last 5, 10, and 15 minutes. This is typically 0.00 on an idle or lightly loaded system.

CPU usage A bar chart and percentage of CPU time in use by the firewall. Note that viewing the dashboard will increase the CPU usage a bit, depending on the platform. On slower platforms this is likely to read significantly higher than it would be otherwise.

Memory usage The current amount of RAM in use by the system. Note that unused RAM is often allocated for caching and other tasks so it is not wasted or idle, so this number may show higher than expected even if it is operating normally.

Swap usage The amount of swap space in use by the system. If the system runs out of physical RAM, and there is swap space available, lesser used pages of memory will be paged out to the swap file on the hard drive. This indicator only shows when the system has swap space configured, which will only be on full installs.

Disk usage The amount of space used on the boot disk or storage media. The type and location of mounted filesystems are shown, including memory disks when present.

Thermal Sensors

The **Thermal Sensors** widget displays the temperature from supported sensors when present. For many popular Intel and AMD-based chips, the sensors may be activated by choosing the appropriate sensor type under **System > Advanced** on the **Miscellaneous** tab under **Thermal Sensors**

A bar is displayed for each sensor, which typically corresponds to each CPU core. The warning and critical thresholds may be configured in the widget settings.

Traffic Graphs

The Traffic Graphs widget contains a live SVG graph for the traffic on each interface. The interfaces displayed are configurable in the widget settings. The default refresh rate of the graphs is once every 10 seconds, but that may also be adjusted in the settings for this widget. The graphs are drawn the same way as those found under Status Traffic Graph.

Wake On LAN

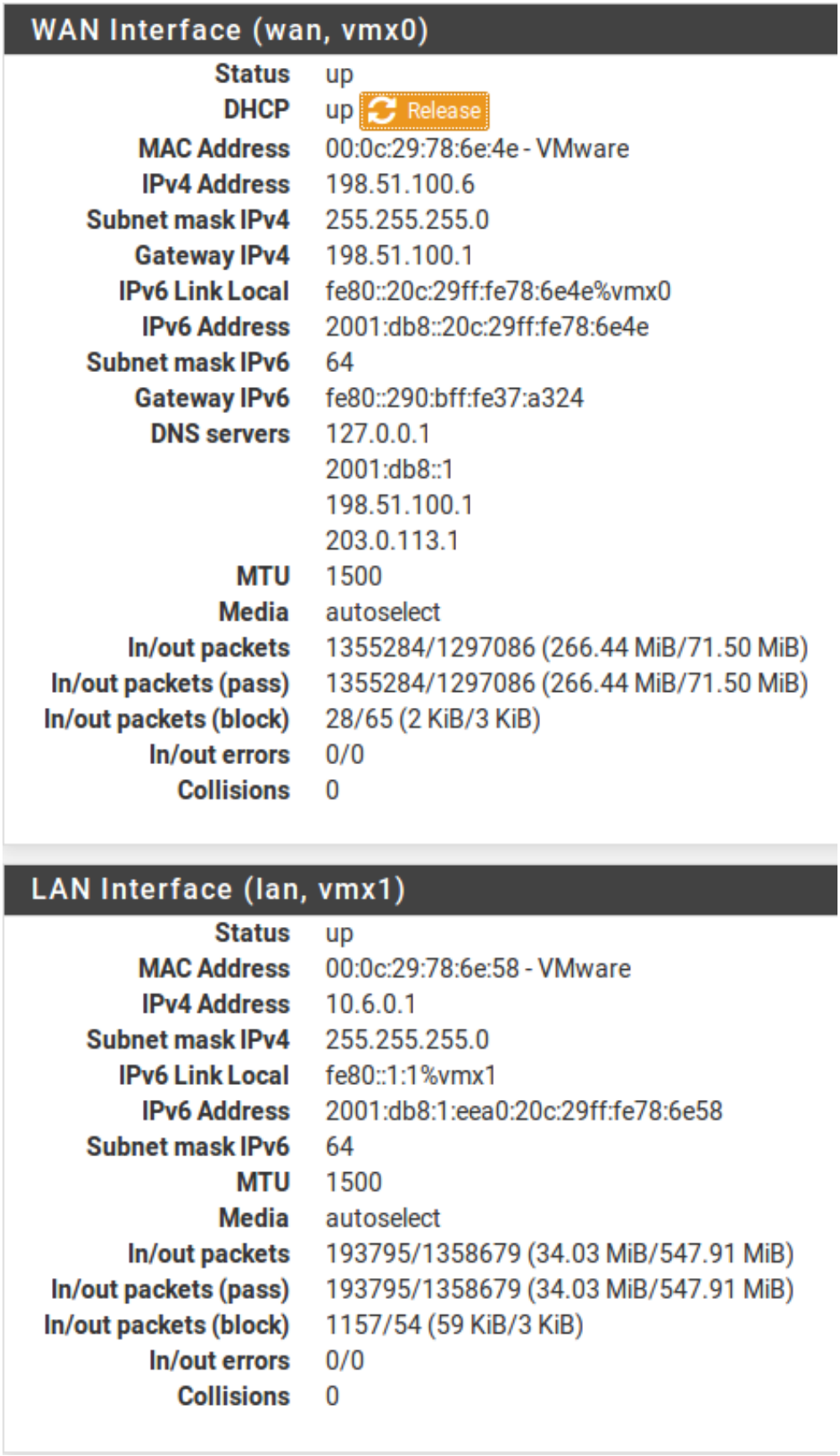
The Wake on LAN widget shows all of the WOL entries configured under Services Wake on LAN, and offers a quick means to send the magic packet to each system in order to wake it up. The current status of a system is also shown. To wake up a system, click  next to its entry.

27.1.2 Interface Status

The status of network interfaces may be viewed at **Status > Interfaces**. In the first part of Figure *Interface Status*, a DHCP WAN connection has been made and the IPv4 and IPv6 address, DNS, etc have been obtained automatically. The MAC address, media type, in/out packets, errors, and collisions for the network interface are all visible. Dynamic connection types like PPPoE and PPTP have a **Disconnect** button when connected and a **Connect** button when offline. Interfaces obtaining an IP address from DHCP have a **Release** button when there is an active lease, and a **Renew** button when there is not.

In the lower part of the image, the LAN connection is visible. Since this is a normal interface with a static IP address, only the usual set of items are shown.

If an interface status indicates “no carrier” then it typically means that the cable is not plugged in or the device on the other end is malfunctioning in some way. If any errors are shown, they are typically physical in nature: cabling or port errors. The most common suspect is cables, and they are easy and cheap to replace. In some circumstances errors and collisions may appear due to a link speed or duplex mismatch. See *Interface Configuration* for more about setting the speed and duplex of an interface.



WAN Interface (wan, vmx0)

Status	up
DHCP	up Release
MAC Address	00:0c:29:78:6e:4e - VMware
IPv4 Address	198.51.100.6
Subnet mask IPv4	255.255.255.0
Gateway IPv4	198.51.100.1
IPv6 Link Local	fe80::20c:29ff:fe78:6e4e%vmx0
IPv6 Address	2001:db8::20c:29ff:fe78:6e4e
Subnet mask IPv6	64
Gateway IPv6	fe80::290:bff:fe37:a324
DNS servers	127.0.0.1 2001:db8::1 198.51.100.1 203.0.113.1
MTU	1500
Media	autoselect
In/out packets	1355284/1297086 (266.44 MiB/71.50 MiB)
In/out packets (pass)	1355284/1297086 (266.44 MiB/71.50 MiB)
In/out packets (block)	28/65 (2 KiB/3 KiB)
In/out errors	0/0
Collisions	0

LAN Interface (lan, vmx1)

Status	up
MAC Address	00:0c:29:78:6e:58 - VMware
IPv4 Address	10.6.0.1
Subnet mask IPv4	255.255.255.0
IPv6 Link Local	fe80::1:1%vmx1
IPv6 Address	2001:db8:1:eea0:20c:29ff:fe78:6e58
Subnet mask IPv6	64
MTU	1500
Media	autoselect
In/out packets	193795/1358679 (34.03 MiB/547.91 MiB)
In/out packets (pass)	193795/1358679 (34.03 MiB/547.91 MiB)
In/out packets (block)	1157/54 (59 KiB/3 KiB)
In/out errors	0/0
Collisions	0

Fig. 4: Interface Status

27.1.3 Service Status

Many system and package services show the status of their daemons at **Status > Services**.

Each service is shown with a name, a description, and the status, as seen in Figure *Services Status*. The status is listed as **Running** or **Stopped**.

Normally, it is not necessary to control services in this manner, but occasionally there are maintenance or troubleshooting reasons for doing so.

From this view, services can be controlled in various ways:

- Click  to restart a running service

---

Note: Some services will stop and start, others reload the configuration. Check the documentation of each service for details.

- Click  to stop a running service
- Click  to start a stopped service

If available, other shortcuts are shown which navigate to pages related to the service. See *Quickly Navigate the GUI with Shortcuts* for information about shortcut icons.

Services

Service	Description	Status	Actions
bsnmpd	SNMP Service	Running	
dhcpd	DHCP Service	Running	
dnsmasq	DNS Forwarder	Running	
dpinger	Gateway Monitoring Daemon	Running	
ipsec	IPsec VPN	Running	
miniupnpd	UPnP Service	Running	
ntpd	NTP clock sync	Running	
openvpn	OpenVPN server: Vendor Remote Access Server	Running	
openvpn	OpenVPN server: Satellite Offices	Running	
openvpn	OpenVPN server: New York Office Site-to-Site	Running	
openvpn	OpenVPN server: Employee Remote Access Server	Running	
radvd	Router Advertisement Daemon	Running	
relayd	Server load balancing daemon	Running	
sshd	Secure Shell Daemon	Running	

Fig. 5: Services Status

27.1.4 System Activity (Top)

The **Diagnostics > System Activity** page displays list of the top active processes running on the firewall. This is equivalent to running the command `top -aSH` at a shell prompt, except the GUI version does not have the CPU usage summary.

Using this view, it is easy to see processes that consume the most CPU power during a time of high load. For example, if the highest entry is an interrupt processing queue for one of the network cards, and the system isn't pushing enough traffic, it could be one sign that the firewall is trying to push more than the hardware can handle in the current configuration. If the top process is a PHP process, it could be that a browser has requested a GUI page that is processing a large amount of data.

Note: Threads that show **idle** in the **COMMAND** column indicate CPU time that is *not* in use (idle). It is normal for these to show 100% if the firewall has little to no load.

27.1.5 pfInfo

The **Diagnostics > pfInfo** page displays statistics and counters for the firewall packet filter which serve as metrics to judge how it is behaving and processing data. The information shown on the page contains items such as:

Bytes In/Out Bytes transferred in and out of the firewall.

Packets In/Out Packets transferred in or out and passed or blocked counters for each direction.

State Table / Source Tracking Table Statistics about the state table and source tracking table (*Firewall States*).

Current Entries The number of entries in the table

Searches How many times the table has been searched and the current rate of searches, which roughly corresponds to the number of packets being passed by the firewall on current open connections.

Inserts The number of new states added to the table, and the rate at which the states are added. A high rate indicates that there are a lot of new connections being made to or through the firewall.

Removals The number of old states being removed from the firewall.

Counters Statistics an counts for various types of special, unusual or badly formatted packets.

Limit Counters Counters that pertain to packets that have reached or exceeded limits configured on firewall rules, such as max states per IP address.

Table Size Limits State table max size, source node table size, frag table size, number of allowed tables, and maximum number of table entries.

State Timers The current configured timeout values for various connection states for TCP, UDP, and other protocols.

Interface Statistics Per-interface packet counters.

27.1.6 S.M.A.R.T. Hard Disk Status

The firewall can monitor the health of hard drives that support Self-Monitoring, Analysis, and Reporting Technology (S.M.A.R.T.). This mechanism is intended to allow drives to test and track their own performance and reliability, with the ultimate goal of identifying a failing drive before it suffers data loss or causes an outage.

Support for S.M.A.R.T. varies by drive and BIOS, but it is fairly well supported in modern ATA hard drives and SSDs. S.M.A.R.T. may need to be enabled in the BIOS and on the drive.

Note: S.M.A.R.T. is not a perfect metric of locating a failed drive; Many drives that have failed still pass a S.M.A.R.T. test, but generally speaking if S.M.A.R.T. does locate a problem, one does exist, so it is useful to identify disk failures.

The **Diagnostics > SMART Status** page obtains and displays information from drives, performs or aborts drive tests, and displays drive logs.

In every section of the page, a **Device** must be selected before choosing an option. This **Device** is the disk to be tested by S.M.A.R.T.

Warning: If a drive is not listed in the **Device** list, it either does not support S.M.A.R.T. or it is connected to a controller that is not supported for this purpose. In the case of RAID controllers, the controller itself may offer similar functionality or reporting via controller-specific utilities in the shell.

Viewing Drive Information

To view information about a drive:

- Navigate to **Diagnostics > SMART Status**
- Locate the **Information** panel on the page
- Select the **Device** to view
- Select the **Info Type**
- Click  **View**

After reviewing the output, click  **Back** to return to the list of options.

The information types are explained in the next subsections.

Info

The **Info** option shows information about the drive itself, including the make, model, serial number, and other technical information about the drive's capabilities, connection, and operation.:

```
Model Family:     Hitachi Travelstar 5K500.B
Device Model:     Hitachi HTS545050B9A300
Serial Number:    090630PB4400XXXXXXXX
LU WWN Device Id: 5 000cca 597XXXXXX
Firmware Version: PB4OC64G
User Capacity:    500,107,862,016 bytes [500 GB]
Sector Size:      512 bytes logical/physical
```

(continues on next page)

(continued from previous page)

```
Rotation Rate:    5400 rpm
Form Factor:      2.5 inches
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   ATA8-ACS T13/1699-D revision 6
SATA Version is:  SATA 2.6, 3.0 Gb/s
Local Time is:    Fri Oct  7 16:31:20 2016 EDT
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

Health

The **Health** option gives a brief pass/fail status of the drive:

```
SMART overall-health self-assessment test result: PASSED
```

SMART Capabilities

The **SMART Capabilities** choice gives a report about features and tests the drive supports, as in this output:

```
General SMART Values:
Offline data collection status:  (0x00) Offline data collection activity
                                        was never started.
                                        Auto Offline Data Collection: Disabled.
Self-test execution status:      (   0) The previous self-test routine completed
                                        without error or no self-test has ever
                                        been run.
Total time to complete Offline
data collection:                (  645) seconds.
Offline data collection
capabilities:                    (0x5b) SMART execute Offline immediate.
                                        Auto Offline data collection on/off support.
                                        Suspend Offline collection upon new
                                        command.
                                        Offline surface scan supported.
                                        Self-test supported.
                                        No Conveyance Self-test supported.
                                        Selective Self-test supported.
SMART capabilities:            (0x0003) Saves SMART data before entering
                                        power-saving mode.
                                        Supports SMART auto save timer.
Error logging capability:        (0x01) Error logging supported.
                                        General Purpose Logging supported.
Short self-test routine
recommended polling time:        (   2) minutes.
Extended self-test routine
recommended polling time:        ( 158) minutes.
SCT capabilities:              (0x003d) SCT Status supported.
                                        SCT Error Recovery Control supported.
                                        SCT Feature Control supported.
                                        SCT Data Table supported.
```

Attributes

The **Attributes** view is the most useful screen in the majoriy of cases, but it can also be one of the trickiest to interpret. There are several values displayed but the number and values vary widely by make and model.

The following output is from a laptop size traditional HDD:

```
=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_
↪FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x000b   099   099   062    Pre-fail  Always       -
↪ 65537
  2 Throughput_Performance  0x0005   100   100   040    Pre-fail  Offline      -
↪ 0
  3 Spin_Up_Time            0x0007   136   136   033    Pre-fail  Always       -
↪ 2
  4 Start_Stop_Count        0x0012   100   100   000    Old_age   Always       -
↪ 96
  5 Reallocated_Sector_Ct   0x0033   100   100   005    Pre-fail  Always       -
↪ 0
  7 Seek_Error_Rate         0x000b   100   100   067    Pre-fail  Always       -
↪ 0
  8 Seek_Time_Performance   0x0005   100   100   040    Pre-fail  Offline      -
↪ 0
  9 Power_On_Hours          0x0012   061   061   000    Old_age   Always       -
↪ 17502
 10 Spin_Retry_Count        0x0013   100   100   060    Pre-fail  Always       -
↪ 0
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -
↪ 96
191 G-Sense_Error_Rate      0x000a   100   100   000    Old_age   Always       -
↪ 0
192 Power-Off_Retract_Count 0x0032   100   100   000    Old_age   Always       -
↪ 37
193 Load_Cycle_Count        0x0012   093   093   000    Old_age   Always       -
↪ 77869
194 Temperature_Celsius     0x0002   152   152   000    Old_age   Always       -
↪ 36 (Min/Max 19/41)
196 Reallocated_Event_Count 0x0032   100   100   000    Old_age   Always       -
↪ 0
197 Current_Pending_Sector  0x0022   100   100   000    Old_age   Always       -
↪ 0
198 Offline_Uncorrectable   0x0008   100   100   000    Old_age   Offline      -
↪ 0
199 UDMA_CRC_Error_Count    0x000a   200   200   000    Old_age   Always       -
↪ 0
223 Load_Retry_Count        0x000a   100   100   000    Old_age   Always       -
↪ 0
```

There is a thorough article on Wikipedia for S.M.A.R.T. that includes a guide for interpreting the values. Some values are more obvious than others, for example the counts for reallocated sectors should be at or near zero. Others can be harder such as the Raw Read Error Rate, which on most drives should be low, but there are Seagate and similar drives that output gibberish or a random high number in that field that makes it useless on those disks.

A few of the values are informational, such as the Start/Stop Count, Power Cycle Count, and Power On Hours which give a sense of the overall age and usage for the drive. A high value isn't necessarily bad for those, but if the drive is

extraordinarily old, or has been power cycled a great many times, then have a plan prepared to replace the disk in the near future. The drive's Temperature can give an indication of its environment, and if the temperature is too high, it can lead to stability issues.

The Load Cycle Count is a special value for spinning disks, since it indicates the number of times the heads have been parked. Some laptop drives will automatically park the heads after a short time, but an OS like pfSense® will want to write periodically, which brings the heads out again. The head parking only makes sense in a mobile device that moves a lot so the heads have less chance of impacting the platter; In a server/firewall situation, it's completely unnecessary. Drives are only capable of 100,000-300,000 load cycles in their lifetime, which means the count gets run through quickly if the heads are continually parked and unparked. pfSense attempts to disable the power management features of hard drives at boot time because otherwise the drive could fail prematurely after running this count up high. This cycling happening is typically audible on drives as a soft clicking noise.

To contrast the above, the following output is from an SSD:

```
=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 10
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_
↪FAILED RAW_VALUE
  5 Reallocated_Sector_Ct   0x0032   100   100   000    Old_age   Always       -
↪ 0
  9 Power_On_Hours_and_Msec 0x0032   100   100   000    Old_age   Always       -
↪ 4198h+12m+31.560s
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always       -
↪ 219
170 Available_Reservd_Space 0x0033   100   100   010    Pre-fail  Always       -
↪ 0
171 Program_Fail_Count      0x0032   100   100   000    Old_age   Always       -
↪ 0
172 Erase_Fail_Count        0x0032   100   100   000    Old_age   Always       -
↪ 0
174 Unexpect_Power_Loss_Ct  0x0032   100   100   000    Old_age   Always       -
↪ 184
183 SATA_Downshift_Count    0x0032   100   100   000    Old_age   Always       -
↪ 0
184 End-to-End_Error        0x0033   100   100   090    Pre-fail  Always       -
↪ 0
187 Uncorrectable_Error_Cnt 0x0032   100   100   050    Old_age   Always       -
↪ 0
190 Airflow_Temperature_Cel 0x0022   053   065   000    Old_age   Always       -
↪ 53 (Min/Max 19/65)
192 Power-Off_Retract_Count 0x0032   100   100   000    Old_age   Always       -
↪ 184
199 UDMA_CRC_Error_Count    0x0032   100   100   000    Old_age   Always       -
↪ 0
225 Host_Writes_32MiB       0x0032   100   100   000    Old_age   Always       -
↪ 21422
226 Workld_Media_Wear_Indic 0x0032   100   100   000    Old_age   Always       -
↪ 65535
227 Workld_Host_Reads_Perc  0x0032   100   100   000    Old_age   Always       -
↪ 17
228 Workload_Minutes        0x0032   100   100   000    Old_age   Always       -
↪ 65535
232 Available_Reservd_Space 0x0033   100   100   010    Pre-fail  Always       -
↪ 0
233 Media_Wearout_Indicator 0x0032   100   100   000    Old_age   Always       -
↪ 0
```

(continues on next page)

(continued from previous page)

```
241 Host_Writes_32MiB       0x0032   100   100   000    Old_age   Always       -
↪ 21422
242 Host_Reads_32MiB        0x0032   100   100   000    Old_age   Always       -
↪ 4635
249 NAND_Writes_1GiB        0x0013   100   100   000    Pre-fail  Always       -
↪ 595
```

The metrics shown for an SSD can be significantly different, as seen above. In particular, SSDs can give an estimate of their remaining lifetime, writes of various sizes, errors rates, write failures, and other SSD-specific values in place of the other values that do not apply to an SSD.

All

Selecting **All** will, as the name implies, show all of the information above and also includes the drive logs and self-test results.

Drive Self-tests

To perform a test on a drive:

- Navigate to **Diagnostics > SMART Status**
- Locate the **Perform Self-Tests** panel on the page
- Select the **Device** to test
- Select the **Test Type**
- Click  **Test**

The types of tests are described in the following subsections.

Offline

An **Offline** test is called so because it is done while the disk is idle. This test can make accessing the drive slow while it is happening, but if there is a lot of disk activity, the drive may delay the test until the disk becomes idle again. Because of this variability, the exact time the test takes is hard to predict. An estimate of the time to complete an offline test for a given disk is shown in the **S.M.A.R.T. Capabilities**. An offline test will also cause the drive to update several of the S.M.A.R.T. attributes to indicate the results. After running a test and checking the results, review the **S.M.A.R.T. Attributes** again as well as the **Error** log.

Short

The **Short** test takes around ten minutes and checks the drive's mechanics and reading performance. A more accurate estimate of the length the test will take on a drive can be seen in the **S.M.A.R.T. Capabilities**. To see the results of this test, view the **Self-test** Logs. It can be run at any time and it does not typically impact performance.

Long

The **Long** test is similar to the **Short** test but is more thorough. The time taken by the test depends on the size of the disk, but it is much longer than the short test on its own. A more accurate estimate of the length the test will take on a drive can be seen in the **S.M.A.R.T. Capabilities**. As with the short test, the results end up in the **Self-test** Logs.

Conveyance

This test is not supported by all drives. Its primary purpose is to test the drive after it has been physically relocated to determine if any components have been damaged by the move. In most cases it only takes a few minutes to complete. To determine if a drive supports a conveyance test, refer to the **S.M.A.R.T. Capabilities** output.

Canceling Active Tests

To cancel an active test on a drive:

- Navigate to **Diagnostics > SMART Status**
- Locate the **Abort** panel on the page
- Select the **Device** currently running a test that must be canceled
- Click  **Abort**

Any active tests on the drive will be stopped.

Drive Logs

The Drive Logs contain information and errors, usually related to self-tests and potentially other errors encountered.

To view drive logs:

- Navigate to **Diagnostics > SMART Status**
- Locate the **View Logs** panel on the page
- Select the **Device** to view
- Select the **Log Type**
- Click  **View**

Error Log

The **Error** log on a drive contains a record of errors encountered during the drive's operation, such as read errors, uncorrectable errors, CRC errors, and so on. Running an **Offline** test will also make the drive print more errors here if they are found during the test.

Self-test Logs

The **Self-test** logs contain a record of several recent self-tests run on the drive. It shows the type of test, the results of the test, and in the case of tests that were stopped prematurely, it shows the percentage of the test remaining.

If an error is encountered during a test, the first logical block address (LBA) is printed to help determine where in the disk the problem lies.

27.1.7 Filter Reload Status

The current status of a filter reload may be viewed in the pfSense® webGUI at **Status > Filter Reload**. A link to this page is available any time a filter change is made. The progress of the reload is displayed automatically, and it is updated automatically.

Normally, updates happen fast enough that `Done.` is the only message shown, indicating that there are no pending changes. With larger configurations, some delays are possible.

Press **Reload Filter** to initiate another filter reload. The progress will be updated as the process progresses.

When configuration synchronization is enabled, a **Force Config Sync** button is also present on this page. That button will trigger a forced synchronization of the firewall configuration by XMLRPC to the node specified by the *current configuration*.

27.1.8 Viewing the Contents of Tables

Aliases and other similar list of addresses are stored in a `pf` structure called a **Table**. These tables can be relatively static, as with the bogons list or aliases, or dynamic for things like snort or IP addresses exceeding connection limits. An alias becomes a "Table" once it has been loaded into the firewall ruleset. Tables may contain both IPv4 and IPv6 addresses, and the appropriate addresses are used based on the rules in which the tables are referenced.

The contents of these tables can be viewed at **Diagnostics > Tables**, which displays system and user-defined tables. On that page, select the desired table from the **Table** drop-down and the firewall will display its contents. If any alias contains a hostname, the contents of the alias are populated from DNS. Viewing the resulting table here confirms which IP addresses are in the table at that moment.

Individual entries may be removed by clicking  at the end of their row. Tables which are defined manually or by a file will be refreshed when the system performs a filter reload, so it is best to edit an alias and remove an entry rather than removing it from this page. Removing entries is best used for dynamic tables to remove an entry before it automatically expires.

Default Tables

The firewall includes several tables by default, depending on which features are enabled:

bogons/bogonsv6 If any interface is configured with *Block Bogon Networks* active, these tables will be present on the firewall. An  **Update** button is also presented for the bogon tables that will immediately re-fetch the bogons data rather than waiting for the usual monthly update.

tonatsubnets When using automatic outbound NAT, this table shows the list of networks for which automatic outbound NAT is being performed. Inspecting the table can aid in diagnosing tricky NAT issues to confirm if a subnet will have automatic outbound NAT applied to its traffic.

snort2c A dynamic table containing blocked offenders from IDS/IPS packages, Snort and Suricata.

virusprot A dynamic table containing addresses that have exceeded defined limits on firewall rules.

webConfiguratorlockout A dynamic table containing clients that repeatedly failed GUI login attempts.

sshlockout Similar to webConfiguratorlockout but used for tracking clients that fail repeated SSH login attempts.

27.1.9 Firewall States

pfSense® is a *stateful firewall* and uses one state to track each connection to and from the firewall. These states may be viewed in several ways in the WebGUI and from the console.

Viewing in the WebGUI

A listing of the firewall state table contents is available in the WebGUI by navigating to **Diagnostics > States**. Figure *Example States* shows a sample of the output displayed by the GUI.

The firewall displays several columns on this page, each with important information:

Interface The interface to which the state is bound. This is the interface through which the packet initially entered or exited the firewall.

Protocol The protocol of the traffic that created the state, such as TCP, UDP, ICMP, or ESP.

Source and Destination This column is in two parts, first the source, then an arrow indicating direction, and then the destination. The source and destination may also have a port number listed if the protocol in question uses ports. In cases where NAT is applied (outbound NAT, port forwards, or 1:1 NAT), the address is shown both before and after NAT has been applied.

For NAT such as outbound NAT which translates the source, the source section displays the translated source, and the original source inside parenthesis. For NAT types that translate the destination, such as port forwards, the destination section shows the translated destination and the original destination in parenthesis.

State The current status of the connection being tracked by this state entry. The specific values vary depending on the protocol. For example, TCP has many more state types than UDP or other connectionless protocols. The entry in this column contains two parts separated by a colon. The first part is the state for the source side, and the second part is the state for the destination side. See *Interpreting States* for more detail.

Packets The number of packets observed matching the state from the source and destination sides.

Bytes The total size of packets observed matching the state from the source and destination sides.

Individual states may be removed by clicking  at the end of their row.

WAN	tcp	198.51.100.6:37246 -> 162.208.119.39:443	TIME_WAIT:TIME_WAIT	91 / 89	6 KiB / 120 KiB	
LAN	tcp	10.6.0.114:49266 -> 52.88.223.32:443	ESTABLISHED:ESTABLISHED	248 / 244	18 KiB / 20 KiB	
WAN	tcp	198.51.100.6:55087 (10.6.0.114:49266) -> 52.88.223.32:443	ESTABLISHED:ESTABLISHED	248 / 244	18 KiB / 20 KiB	
WAN2	icmp	203.0.113.106:36339 -> 203.0.113.1:36339	0:0	832.827 K / 921	22.26 MiB / 25 KiB	

Fig. 6: Example States

See also:

Firewall Maximum States

Filtering States

The **State Filter** panel enables quick searching of the state table contents to find items of interest.

To search for a state:

- Select a specific **Interface** in the **State Filter** panel or leave it on *all* to match all interfaces.
- Enter a **Filter Expression** which is a simple string of text to match exactly in the entry. Regular expressions are not supported in this field.
- Click  **Filter** to locate the results.

All columns are searched for matching text, and only entries matching the text are displayed.

Tip: Searching for an IP address or subnet will also present a  **Kill States** button which, when clicked, will remove all states originating from or going to the entered IP address or subnet.

Interpreting States

The **State** column for each state table entry provides information necessary to determine exactly what is happening with the connection. Each state entry contains two values with a colon between them, marking which value represents the state of the source (left), and which represents the destination (right).

A few of the most common state types are:

SYN_SENT For TCP connections, this indicates that the side showing this state sent a TCP SYN packet attempting to start a connection handshake.

CLOSED For TCP connections, the side with this status considers the connection closed, or no traffic has been received.

ESTABLISHED A TCP connection is considered fully established by this side.

TIME_WAIT/FIN_WAIT A TCP connection is in the process of closing and finishing up.

NO_TRAFFIC No packets have been received that match the state from this side.

SINGLE A single packet has been observed on this state from this side.

MULTIPLE Multiple packets have been observed on this state from this side.

Common pairings frequently found in the state table include:

ESTABLISHED:ESTABLISHED A fully established two-way TCP connection.

SYN_SENT:CLOSED The side showing *SYN_SENT* has sent a TCP SYN packet but no response has been received from the far side. Often this is due to the packet not reaching its destination, or being blocked along the way.

SINGLE:NO_TRAFFIC Similar to the above, but for UDP and other connectionless protocols. No response has been received from the destination side.

SINGLE:MULTIPLE For UDP and other connectionless protocols, commonly observed with DNS where the client sends one packet but receives a large response in multiple packets.

MULTIPLE:MULTIPLE For UDP and other connectionless protocols, there are multiple packets in both directions, which is normal for a fully operational UDP connection.

0:0 Indicates that there is no state level data. Typically only found on ICMP states, since ICMP does not have state levels like other protocols.

States Summary

The **State Table Summary**, accessible from **Diagnostics > States Summary**, provides statistics generated by an in-depth analysis of the state table and the connections therein.

The report includes the IP address, a total state count, and breakdowns by protocol and source/destination ports. Hovering over the ports shows a tooltip display of the full port list instead of the total number of ports. Depending on the firewall environment, high values by any metric may be normal.

The report includes the following categories:

By Source IP Address States summarized by the source IP address. This is useful for finding a potential source of attack, or a port scan or similar type probe/attack.

By Destination IP Address States summarized by the destination IP address of the connection. Useful for finding the target of an attack or identifying servers.

Total per IP Address States summarized by all connections to or from an IP address. Useful for finding active hosts using lots of ports, such as bittorrent clients.

By IP Address Pair Summarizes states between two IP addresses involved in active connections. Useful for finding specific client/server pairs that have unusually high numbers of connections.

Warning: The **States Summary** can take a long time to process and display, especially if the firewall has an exceptionally large state table or a slow processor. In cases where the state table is extremely large, the page may not display properly or the page may fail with a memory error. In these cases, the summary page cannot be used.

Source Tracking States

When using **Sticky Connections**, the firewall maintains a source tracking table that records mappings of internal IP addresses to specific external gateways for connections that were passed by a rule utilizing a Load Balancing gateway group (Multiple gateways on the same tier). By default these associations only exist so long as there are active states from the internal IP address. There is a configurable timeout for these source tracking entries to allow them to exist longer if necessary.

See also:

For additional information about Sticky Connections and their related options, see *Sticky Connections*.

The source tracking associations are shown on **Diagnostics > States** on the **Source Tracking** tab, which is only visible if Sticky Connections are enabled.

The Source Tracking page lists the following information:

Source-to-Destination The mapping of a local IP address to a specific load balanced gateway.

 # States The number of states matching this source IP address to any destination, including traffic that is not load balanced.

 # Connections The number of states matching this source IP address which utilize the gateway. For example, connections leaving from this source to an Internet host.

Rate The rate of packets matching this source tracking entry.

These associations can be individually removed by clicking the **Remove** button at the end of each row.

Reset State Table / Source Tracking Table

Certain situations call for resetting the state table to force all existing connections to close and reestablish. The most notable examples are making changes to NAT rules, firewall block rules, or traffic shaping. When these types of changes are made, resetting the state table is the only way to make sure all connections respect the new ruleset or traffic shaping queues.

Warning: Resetting the state table is disruptive, but clients may immediately reconnect provided they are still passed by the current firewall rules.

Both the state table and the source tracking table may be reset from **Diagnostics > States** on the **Reset States** tab. To reset the tables, check either **State Table**, **Source Tracking**, or both, and then click  **Reset**.

Warning: The browser will appear to lose connection with the firewall when resetting the state table. Once the browser realizes the old connection is invalid, it will reconnect. Close and reopen the browser to reconnect faster.

Viewing States with pfTop

pfTop is available from the GUI and the system console menu, and offers live views of the firewall ruleset, state table information, and related statistics.

pfTop in the GUI

In the GUI, pfTop can be found at **Diagnostics > pfTop**. The GUI offers several options to control the output:

View Controls the type of output displayed by pfTop. Not all views will contain meaningful information for every firewall configuration.

Default Shows a balanced amount of information, based around the source and destination of the traffic.

Label Centered around firewall rule descriptions.

Long Similar to the default view, but tailored for wider displays with longer rows for more columns of information. Shows the gateway after the destination.

Queue Shows the ALTQ traffic shaping queues and their usage.

Rules Shows firewall rules and their usage.

Size Shows states that have passed the most data.

Speed Shows states that have high-rate traffic.

State Shows status of states.

Time Shows long-lived states.

Sort By Some views can be sorted. When sorting is possible, the following sort methods are available. When selected, the view is sorted by the chosen column in descending order:

None No sorting, the natural order shown by the chosen view.

Age The age of the states.

Bytes The amount of data sent matching states.

Destination Address The destination IP address of the state.

Destination Port The destination port number of the state.

Expiry The expiration time of the state. This is the countdown timer until the state will be removed if no more data matches the state.

Peak The peak rate of traffic matching a state in packets per second.

Packet The number of packets transferred matching a state.

Rate The current rate of traffic matching a state in packets per second.

Size The total amount of traffic that has matched a state.

Source Port The source port number of the state.

Source Address The source IP address of the state.

Maximum # of States On views that support sorting, this option limits the number of state entries shown on the page.

pfTop on the Console

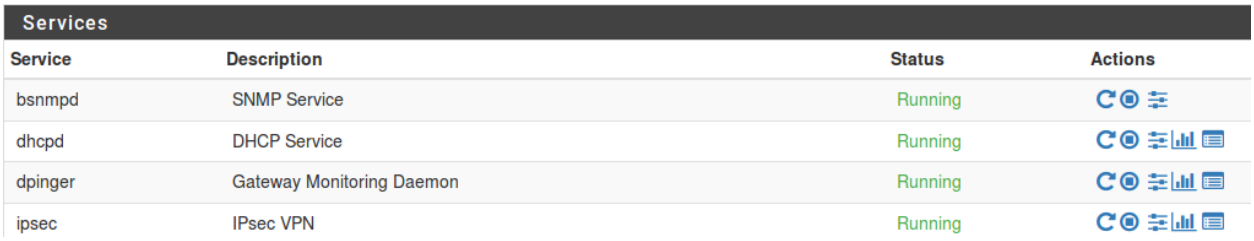
To access pfTop from the console or via ssh, use option `9` from the menu or run `pftop` from a shell prompt.

While viewing pfTop in this way, there are several methods to alter the view while watching its output. Press `h` to see a help screen that explains the available choices. The most common uses are using `0` through `8` to select different views, `space` for an immediate update, and `q` to quit. See the previous section for details on the meaning of the available views and sort orders.

The output is dynamically sized to the terminal width, with wider terminals showing much more information in additional columns.

27.1.10 Status

The status of the DHCP server service itself is at **Status > Services**. If the DHCP server is enabled, its status will be shown as **Running**, as in Figure *DHCP Daemon Service Status*. The buttons on the right side allow restarting or stopping the DHCP server daemon. Restarting is not normally necessary as pfSense® will automatically restart the service when configuration changes are made that require a restart. Stopping the service is also likely not necessary, as the service will stop when all instances of the DHCP server are disabled.



Services

Service	Description	Status	Actions
bsnmpd	SNMP Service	Running	
dhcpd	DHCP Service	Running	
dpinger	Gateway Monitoring Daemon	Running	
ipsec	IPsec VPN	Running	

Fig. 7: DHCP Daemon Service Status

27.1.11 Leases

The currently assigned DHCP leases are viewable at **Status > DHCP leases**. This page shows various aspects of the client leases. These include:

- The assigned IP address
- The client MAC address
- The hostname (if any) that the client sent as part of the DHCP request
- The description for a host with a DHCP static mapping
- The beginning and end times of the lease
- Whether or not the machine is currently online (in the ARP table)
- Whether or not the lease is active, expired, or a static registration

View inactive leases

By default, only active and static leases are shown, but everything, including the expired leases, may be displayed by clicking **Show all configured leases**. To reduce the view back to normal, click **Show active and static leases only**.

Wake on LAN Integration

Clicking the  icon to the right of the lease sends a Wake on LAN (WOL) packet to that host. Click  to create a WOL entry for the MAC address instead. For more details about Wake on LAN, see *Wake on LAN*.

Add static mapping

To create a static mapping from a dynamic lease, click  the to the right of the lease. This will pre-fill the MAC address of that host into the **Edit static mapping** screen. Add the desired IP address, hostname and description and click **Save**.

Delete a lease

While viewing the leases, an inactive or expired lease may be manually deleted by clicking  at the end of its line. This option is not available for active or static leases, only for offline or expired leases.

27.1.12 DHCP Service Logs

The DHCP daemon will log its activity to **Status > System Logs**, on the **DHCP** tab. Each DHCP request and response will be displayed, along with other status and error messages.

27.1.13 Viewing DHCPv6 Leases

A list of active and inactive **DHCPv6 Leases** (DHCP leases for IPv6 hosts) and delegated prefixes can be viewed in pfSense® software by navigating to **Status > DHCPv6 Leases**.

All active leases are shown, along with the IPv6 address, IAID, DUID, MAC address, hostname, lease start and end times, lease type, and whether or not the system is online. (As with the *NDP Table*, this is not always a reliable indicator)

IPv6 hosts are identified by a combination of the Interface Association Identifier (IAID) and the DHCP Unique Identifier (DUID). The DUID is meant to be unique *per device* and the IAID is unique *per interface on the device*. Due to variances in DHCPv6 clients between operating systems and manufacturers some clients do not send an IAID, and some DUID values are sized differently than others.

To view expired leases, click **Show All Configured Leases**. To switch the view back, click **Show Active and Static Leases**.

A DHCPv6 static IP mapping may be added by clicking  . An offline dynamic lease may be deleted by clicking  .

Delegated Prefixes

When **Prefix Delegation** is enabled, the bottom of the page lists delegated prefixes and their routing. Similar to the lease status, the DUID of the target system is shown along with the start and end time of the delegation. The **IPv6 Prefix** column shows the delegated prefixes, their prefix length, and the address to which they are routed. The target address and DUID will match one of the leases in the list at the top of the screen.

27.1.14 Gateway Status

In the pfSense® webGUI, **Status > Gateways** displays the current status of all configured gateways.

The status output includes the gateway name, gateway IP, Monitor IP, status and description. The status field will show Online, Offline, or Warning, and a textual description of the problem (or success).

The **Gateway Groups** tab shows the status of gateway group members and the groups as a whole.

27.1.15 CARP Status

The CARP status page located through the pfSense® webGUI at **Status > CARP (failover)** shows the current status of all configured CARP *Virtual IP addresses*. It also provides some controls to enable and disable CARP for troubleshooting and maintenance.

For each VIP, the **Interface**, **Virtual IP**, and **Status** are shown.

The **Interface** column shows the interface name or identifier for the VIP at the operating system level. This has changed over the years, and currently shows as "_vip" (ex: *lan_vip1*) on pfSense software version 2.1.x, and on pfSense software version 2.2, it shows as "@" (ex: *LAN@1*).

The **Virtual IP** column shows the IP address associated with the entry.

The **Status** column shows **MASTER**, **BACKUP**, or **INIT**, which are:

- **MASTER**: Indicates this node is accepting all traffic for this VIP
- **BACKUP**: Indicates this node is monitoring CARP advertisements and not accepting traffic for the VIP.
- **INIT**: Generally indicates a problem with the VIP. Either the VIP is not configured at the OS level, or the interface upon which it is configured is down or has a problem.

A toggle button at the top of the list will show either **Disable CARP** or **Enable CARP** depending on the current status. Disabling CARP will remove the VIPs from the system, and the next available node will take over as **MASTER** for the CARP VIPs. This setting is not remembered across reboots.

On the primary node, each VIP should show **MASTER**. On the secondary node, each VIP should show **BACKUP**. If both nodes show **MASTER**, there is usually a problem at layer 2 (the switch) preventing the two nodes from seeing each others' advertisements. See *CARP Configuration Troubleshooting*.

Maintenance Mode

There is a toggle button to **Enter Persistent CARP Maintenance Mode** or **Leave Persistent CARP Maintenance Mode**. This mode persists across reboots, so it is useful for performing maintenance or upgrades on the primary node such that it does not cause it to take back over prematurely before it is ready.

Widget

This is also a **CARP Status** widget available for the *Dashboard* that shows the same information in a condensed format without the control buttons.

27.1.16 Captive Portal Status

The **Captive Portal Status** page is available through the pfSense® software webGUI at **Status > Captive Portal**.

To view the status for a Zone, select it from the **Captive Portal Zone** drop-down.

This page displays a list of online users, including their **IP address**, **MAC address**, **Username**, and **Session Start Time**.

The exact output for each field depends on the portal configuration. For example, if MAC address filtering is disabled, the MAC address may not be displayed. If user authentication is not required, then there will be no **Username** to display.

One of several different user styles may appear in the list.

For a portal with **No Authentication**, a listing like Figure *Online Captive Portal Users No Authentication* is shown.

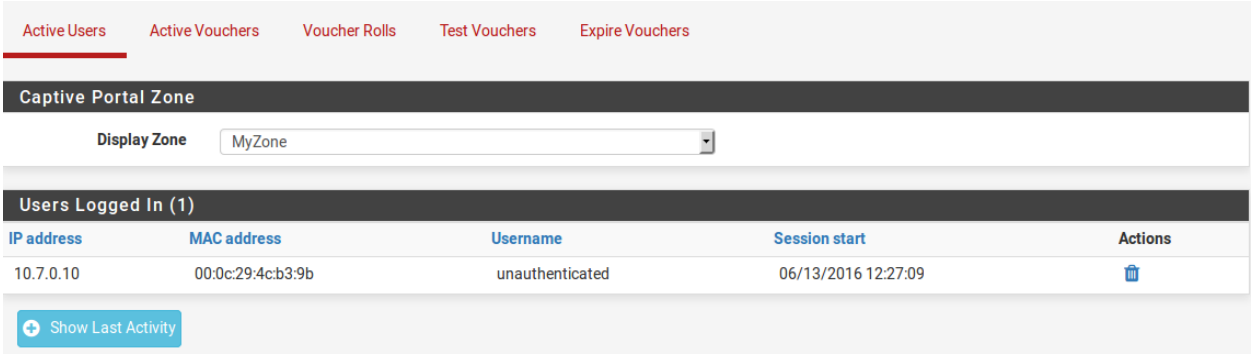


Fig. 8: Online Captive Portal Users No Authentication

For **Local User Manager** or **RADIUS** authentication is selected, a listing like Figure *Online Captive Portal Users User Authentication* is shown. If **RADIUS** with **MAC Authentication** is used, the username will be the MAC address.



Fig. 9: Online Captive Portal Users User Authentication

If vouchers are active, users that signed on using a voucher will show like Figure *Online Captive Portal Users Vouchers*.



Fig. 10: Online Captive Portal Users Vouchers

27.1.17 Viewing Active Network Sockets

The **Diagnostics > Sockets** page presents a list of active TCP/IP sockets for both IPv4 and IPv6 *used by the firewall itself*.

This list is useful for determining which IP addresses and ports are in use by various system processes or packages. It is interpreted from the output of the FreeBSD command "*sockstat*".

By default, only *listening* sockets are shown. Click **Show all socket connections** to also display sockets in use by the system making connections to external hosts.

The output of this command only shows sockets used by the firewall OS for daemons or other programs on the firewall. It does **not** show connections for traffic passing through the firewall.

27.1.18 ARP Table

ARP (Address Resolution Protocol) is used for locating IPv4 systems on a local network by MAC address.

The ARP table in pfSense® software displays a list of systems on the network that have attempted to talk to or through the pfSense firewall within the past few minutes. If a system is up but has not talked to (or through) the pfSense firewall it will not show up in the ARP table.

To view the list of systems currently seen by pfSense software, click **Diagnostics > ARP Table**. This list shows the **IP Address**, **MAC Address**, **Hostname** and the **Interface** where each system was last seen.

For IPv6 hosts, see *NDP Table*.

27.1.19 NDP Table

The **NDP Table**, found under the **Diagnostics** menu in the pfSense® webGUI, shows the IPv6 Neighbor Discovery Protocol list. This list contains all of the current IPv6 peers, and is roughly analogous to the *ARP Table* for IPv4.

The NDP table shows the peer's IPv6 address, its MAC address, the Hostname (if known), and the interface upon which that peer IP address was last observed.

As with ARP, these are only the hosts that are actively talking to or through the firewall, and is not a perfect indicator of a host's actual online status. If a host is in the table, it is likely online, but if it is not in the list, it does not necessarily mean that it is offline, it may simply not have attempted external communication recently.

27.1.20 Hardware Monitoring Support

FreeBSD, and thus pfSense® software, supports hardware monitoring on a few chipsets. Unfortunately, support for this is limited but growing as hardware is replaced by newer Intel and AMD CPUs that include better monitoring. Where supported, it can be handy.

An Intel or AMD temperature monitor module may be selected under **System > Advanced** on the **Miscellaneous** tab. These work with Intel Core series and later Intel chips, and similarly recent AMD chips, respectively. Support is not universal, but it is common, especially on Intel chips, even Atom-based chips. The temperature can be observed using the Thermal Sensors dashboard widget, or by sysctl:

```
# sysctl -a | grep "dev.cpu.*.temperature"
dev.cpu.0.temperature: 46.0C
dev.cpu.1.temperature: 47.0C
dev.cpu.2.temperature: 47.0C
dev.cpu.3.temperature: 47.0C
```

These are typically on-die sensors so they only represent the CPU core temperatures, not other zones in the system.

FreeBSD also handles some settings through ACPI. To see if the hardware supports temperature monitoring, try the following command from a shell prompt or **Diagnostics > Command**:

```
sysctl hw.acpi.thermal
```

If the hardware is supported, output similar to the following will be shown:

```
hw.acpi.thermal.min_runtime: 0
hw.acpi.thermal.polling_rate: 10
hw.acpi.thermal.user_override: 0
hw.acpi.thermal.tz0.temperature: 22.5C
hw.acpi.thermal.tz0.active: -1
```

(continues on next page)

(continued from previous page)

```
hw.acpi.thermal.tz0.passive_cooling: 1
hw.acpi.thermal.tz0.thermal_flags: 0
hw.acpi.thermal.tz0._PSV: 85.0C
hw.acpi.thermal.tz0._HOT: -1
hw.acpi.thermal.tz0._CRT: 100.0C
hw.acpi.thermal.tz0._ACx: 85.0C -1 -1 -1 -1 -1 -1 -1 -1 -1
```

In this example, there is only one Thermal Zone, and its temperature is 22.5C (72.5F).

From acpi_thermal(4):

```
hw.acpi.thermal.min_runtime
     Number of seconds to continue active cooling once started.  A new
     active cooling level will not be selected until this interval
     expires.

hw.acpi.thermal.polling_rate
     Number of seconds between polling the current temperature.

hw.acpi.thermal.user_override
     If set to 1, allow user override of various setpoints (below).
     The original values for these settings are obtained from the BIOS
     and system overheating and possible damage could occur if
     changed.  Default is 0 (no override).

hw.acpi.thermal.tz%d.active
     Current active cooling system state.  If this is non-negative,
     the appropriate _AC%d object is running.  Set this value to the
     desired active cooling level to force the corresponding fan
     object to the appropriate level.

hw.acpi.thermal.tz%d.passive_cooling
     If set to 1, passive cooling is enabled.  It does cooling without
     fans using cpufreq(4) as the mechanism for controlling CPU speed.
     Default is enabled for tz0 where it is available.

hw.acpi.thermal.tz%d.thermal_flags
     Current thermal zone status.  These are bit-masked values.

hw.acpi.thermal.tz%d.temperature
     Current temperature for this zone.

hw.acpi.thermal.tz%d._PSV
     Temperature to start passive cooling by throttling down CPU, etc.
     This value can be overridden by the user.

hw.acpi.thermal.tz%d._HOT
     Temperature to start critical suspend to disk (S4).  This value
     can be overridden by the user.

hw.acpi.thermal.tz%d._CRT
     Temperature to start critical shutdown (S5).  This value can be
     overridden by the user.

hw.acpi.thermal.tz%d._ACx
     Temperatures at which to switch to the corresponding active cooling
     level.  The lower the _ACx value, the higher the cooling power.
```

(continues on next page)

(continued from previous page)

```
All temperatures are printed in Celsius.  Values can be set in Celsius
(by providing a trailing "C") or Kelvin (by leaving off any trailing letter).
When setting a value by sysctl(8), do not specify a trailing decimal (i.e.,
90C instead of 90.0C).
```

The defaults for these values are taken from the BIOS, and some systems will not allow changes. The only way to know is to try. Before attempting to alter any of these values, set this OID:

```
sysctl -w hw.acpi.thermal.user_override=1
```

These values may be set by adding the appropriate lines for the OIDs to **System > Advanced** on the **Tunables** tab. Try setting them on the fly like so:

```
sysctl -w hw.acpi.thermal.tz0._CRT=120C
```

27.1.21 IPsec Status

The status of all IPsec tunnels may be viewed by browsing to **Status > IPsec**. This page is divided into four tabs.

Overview Tab

This tab lists all enabled IPsec tunnels, the local and remote IP addresses, local and remote networks, tunnel description, and status.

A green icon indicates that the tunnel is up (has SAD and SPD entries, signifying a complete phase 1 and 2 connection).

A yellow icon indicates that the tunnel is not fully up and active.

SAD Tab

Shows the contents of the IPsec Security Association Database. There should be one for each "direction" between *public peer addresses* of an active IPsec tunnel.

SPD Tab

Shows the contents of the IPsec Security Policy Database. There should be one for each direction between *private networks* of an active IPsec tunnel.

Logs Tab

A shortcut to the *IPsec Logs* normally found at **Status > System Logs**, **IPsec** tab.

See Also

- *IPsec Troubleshooting*

27.1.22 Checking the Status of OpenVPN Clients and Servers

The OpenVPN status page at **Status > OpenVPN** shows the status of each OpenVPN server and client. Service start/stop controls are also available for each separate server and client instance on the status page.

For OpenVPN servers in SSL/TLS server mode, the status provides a list of connected remote clients along with their usernames or certificate common names, as seen in Figure *OpenVPN Status for SSL/TLS Server With One Connected Client*. Clients may also be disconnected from this screen by clicking the  at the end of the client row. For these servers a  **Show Routing Table** button is also displayed. Clicking this button will show a table of networks and IP addresses connected through each client certificate.

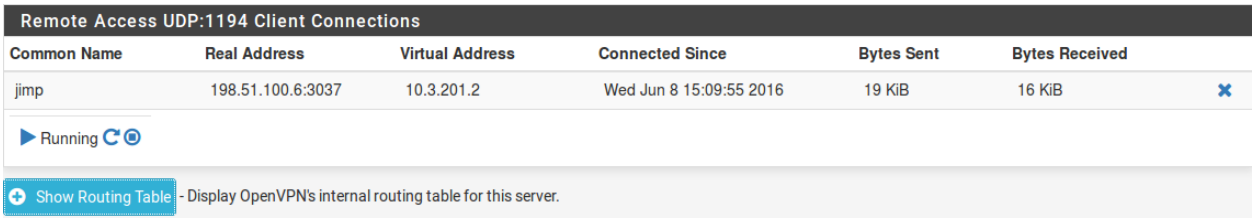


Fig. 11: OpenVPN Status for SSL/TLS Server With One Connected Client

For OpenVPN servers in shared key mode, the status will indicate whether it's running and waiting on connections, or if the remote client has connected.

For OpenVPN clients, the status indicates whether a connection is pending or active.

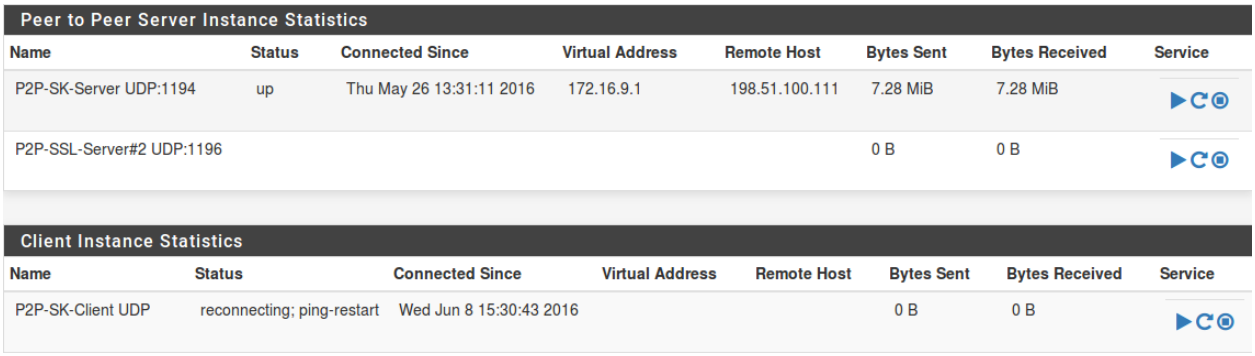


Fig. 12: OpenVPN Status Showing a Server that is up, one waiting for a Connection, and a Client Attempting to Reconnect

27.1.23 Route Table Contents

View the firewall route table at **Diagnostics > Routes**.

Each line shows the destination network, gateway by which it can be reached, route flags, number of references and times the route has been used, the MTU, the interface through which the routed traffic will be sent by the firewall.

The list of routes supports pagination and filtering to aid with viewing large routing tables such as those found with a full BGP feed.

Resolve Names When checked, the firewall attempts a DNS lookup to show hostnames rather than IP addresses for route table entries. This will cause a delay and performance penalty.

Rows to display By default the page displays `100` rows. Choose a new value to show more or less rows.

Filter Search the route table for entries matching this string. The field supports regular expressions for advanced filtering.

Click  **Update** to redisplay the routing table with the current settings.

The meaning of the **Flags** column entries are explained in *Route Table Flags*

Table 1: Route Table Flags

Letter	Flag	Meaning
1	RTF_PROTO1	Protocol specific routing flag #1
2	RTF_PROTO2	Protocol specific routing flag #2
3	RTF_PROTO3	Protocol specific routing flag #3
B	RTF_BLACKHOLE	Just discard pkts (during updates)
b	RTF_BROADCAST	The route represents a broadcast address
C	RTF_CLONING	Generate new routes on use
c	RTF_PRCLONING	Protocol-specified generate new routes on use
D	RTF_DYNAMIC	Created dynamically (by redirect)
G	RTF_GATEWAY	Destination requires forwarding by intermediary
H	RTF_HOST	Host entry (net otherwise)
L	RTF_LLINFO	Valid protocol to link address translation
M	RTF_MODIFIED	Modified dynamically (by redirect)
R	RTF_REJECT	Host or net unreachable
S	RTF_STATIC	Manually added
U	RTF_UP	Route usable
W	RTF_WASCLONED	Route was generated as a result of cloning
X	RTF_XRESOLVE	External daemon translates proto to link address

27.1.24 Wireless Status

The Wireless Status screen (**Status > Wireless Status**) shows a list of access points within range of the wireless radio, along with various statistics about the access points detected.

If a wireless card is acting as an access point, a list of associated clients is displayed along with related information.

This page is only available when a wireless interface is detected in the firewall.

Access Point/Ad-Hoc Peer Capabilities

Name	Description
WME	Wireless Multimedia Extensions (QoS)
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup
RSN	802.11i - Robust Secure Network
HTCAP	802.11n - High Throughput (HT)
ATH	Atheros protocol extensions
VEN	Unknown vendor-specific extensions

Access Point/Ad-Hoc Peer Flags

Flag	Description
E	ESS Mode (Access Point)
I	IBSS Mode (Infrastructure/Client)
c	CF Pollable
C	CF Poll Request
P	Privacy
S	Short Preamble
B	PBCC
A	Channel Agility
s	Short Slot Time
R	RSN
D	DSSSOFDM

Peer Status Flags

Flag	Description
A	Authorized for Data
Q	QoS (WME)
E	Extended Rate (ERP), 802.11g
P	Power Save Mode
H	High Throughput (HT)
H+	HT Compat mode (Setup with vendor OUIs)
W	Wi-Fi Protected Setup (WPS)
N	Transitional Security Network (TSN) association
T	Aggregated MAC Protocol Data Unit (AMPDU) Transmit
R	AMPDU Receive
M	MIMO Power Save
M+	MIMO Power Save with RTS
I	Reduced Interframe Space (RIFS)
S	Short GI in HT40
S+	Short GI in HT40 and HT20
s	Short GI in HT20
t	Aggregated MAC Service Data Unit (AMSDU) Transmit
r	AMSDU Receive

27.1.25 Monitoring the Queues

Monitor the shaper using **Status > Queues** to ensure that traffic shaping is working as intended. As can be seen in Figure *Basic WAN Queues*, this screen shows each queue listed by name, its current usage, and other related statistics.

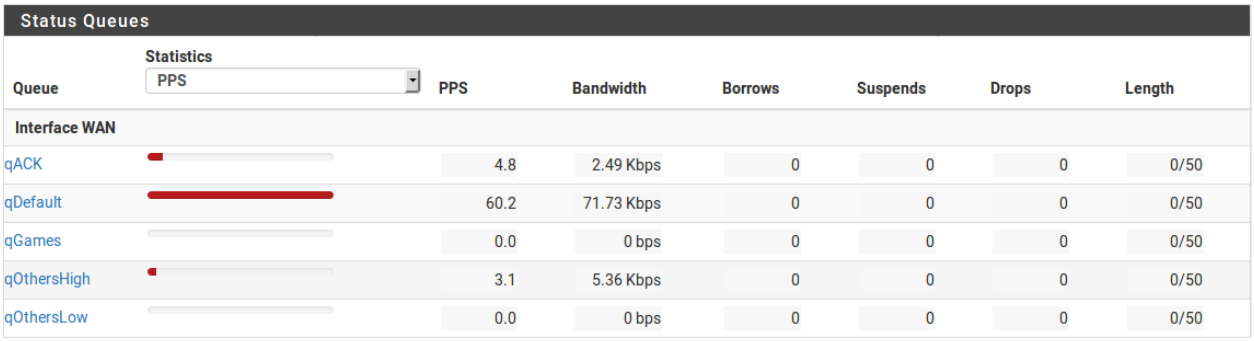
Queue The name of the traffic shaper queue.

Statistics A graphical bar which shows how "full" this queue is.

PPS The rate of queued data in packets per second (PPS)

Bandwidth The rate of queued data in bits per second (e.g. Mbps, Kbps, bps).

Borrows Borrows happen when a neighboring queue is not full and capacity is borrowed from there.



Status Queues

Statistics: PPS

Queue	PPS	Bandwidth	Borrows	Suspends	Drops	Length
Interface WAN						
qACK	4.8	2.49 Kbps	0	0	0	0/50
qDefault	60.2	71.73 Kbps	0	0	0	0/50
qGames	0.0	0 bps	0	0	0	0/50
qOthersHigh	3.1	5.36 Kbps	0	0	0	0/50
qOthersLow	0.0	0 bps	0	0	0	0/50

Fig. 13: Basic WAN Queues

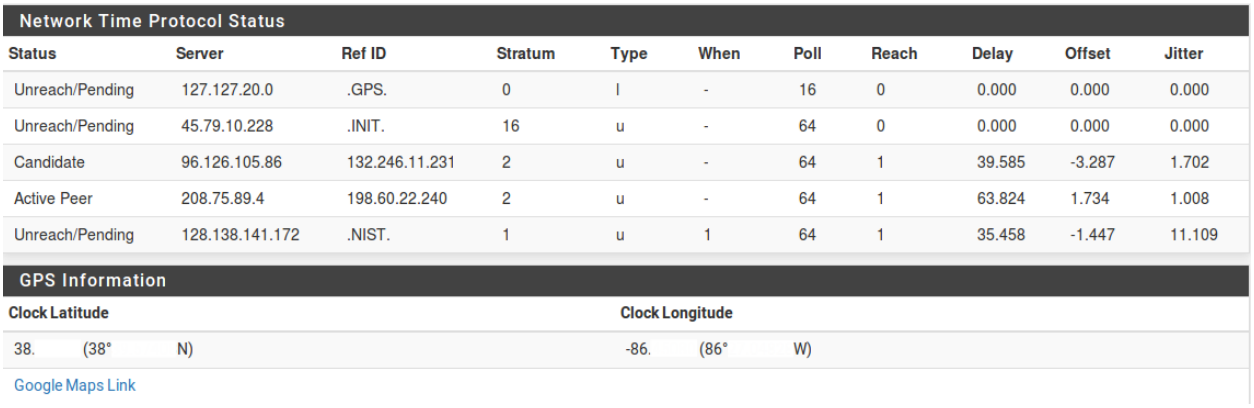
Suspends The suspends counter indicates when a delay action happens. The suspends counter is only used with the CBQ scheduler and should be zero when other schedulers are in use.

Drops Drops happen when traffic in a queue is dropped in favor of higher priority traffic. Drops are normal and this does not mean that a full connection is dropped, only a packet. Usually, one side of the connection will see that a packet was missed and then resend, often slowing down in the process to avoid future drops.

Length The number of packets in the queue waiting to be transmitted, over the total size of the queue.

27.1.26 Status

The NTP status page shows the status of each NTP peer server. This status page can be found at **Status > NTP**. An example of the status is shown in Figure *NTP Daemon Status With GPS Output*.



Network Time Protocol Status

Status	Server	Ref ID	Stratum	Type	When	Poll	Reach	Delay	Offset	Jitter
Unreach/Pending	127.127.20.0	.GPS.	0	l	-	16	0	0.000	0.000	0.000
Unreach/Pending	45.79.10.228	.INIT.	16	u	-	64	0	0.000	0.000	0.000
Candidate	96.126.105.86	132.246.11.231	2	u	-	64	1	39.585	-3.287	1.702
Active Peer	208.75.89.4	198.60.22.240	2	u	-	64	1	63.824	1.734	1.008
Unreach/Pending	128.138.141.172	.NIST.	1	u	1	64	1	35.458	-1.447	11.109

GPS Information

Clock Latitude			Clock Longitude		
38.	(38°	N)	-86.	(86°	W)

Google Maps Link

Fig. 14: NTP Daemon Status With GPS Output

The status screen contains one line for every peer, and lists the peer IP address or server ID, the reference clock ID for the peer and various other values that indicate the general quality of the NTP server from the perspective of this firewall. The first column is the most useful, as it indicates which peer is currently the active peer for time sync, which servers are potential candidates to be peers, and which servers have been rejected and why.

If a serial GPS is connected and configured, the coordinates reported by the GPS device are also listed, along with a link to the coordinates on Google Maps.

Note: The quality of GPS data can vary widely depending on the signal level, the GPS device, and how it is connected. Traditional serial ports are higher quality and better suited to GPS clock usage. USB serial GPS units may

be acceptable, but due to how USB functions, the timing of signals cannot be guaranteed the way it can be with a traditional hard-wired serial port.

27.2 Graphs

These articles cover graphs for monitoring pfSense itself as well as for traffic on interfaces and using additional packages for more detailed monitoring of user throughput/usage.

27.2.1 Monitoring Graphs

pfSense® software has many built-in graphs that monitor different aspects of the system, and they work out-of-the-box with no intervention.

The firewall collects and maintains data about how the system performs, and then stores this data in Round-Robin Database (RRD) files. Graphs created from this data are available under **Status > Monitoring**.

These graphs measure things such as CPU usage, memory usage, state table usage, throughput (in bytes as well as packets), link quality, traffic shaping queue usage, and more.

The graph on that page can be configured to show items from several categories, and a category and graph may be chosen for both the left axis and right axis for easy comparison.

Working with Graphs

The firewall displays a graph showing its CPU usage by default. To view other graphs or to add a second category on another axis, the graph settings must be changed as described in the next section, *Graph Settings*.

Inside the graph, the labels in the top left corner note the sources for the data in the left axis and right axis.

The graph contains a legend at the top right with each of the data sources plotted on the graph. Clicking a data source in the legend will hide it from view.

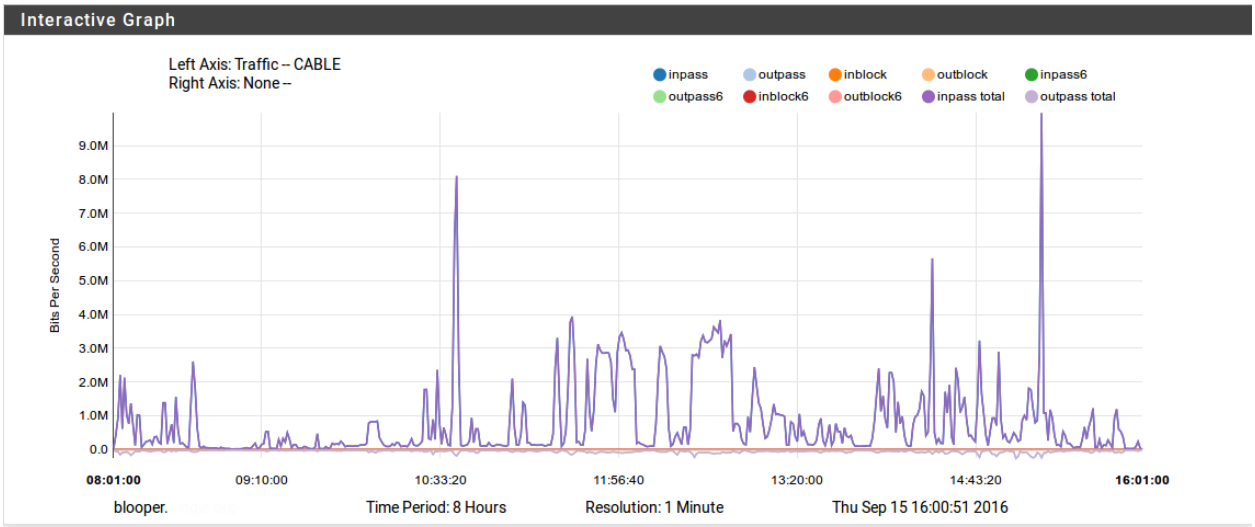
Tip: If a data source has a large spike, click its name in the legend to remove it from the graph. With the larger data source removed, more detail from the other remaining sources will be visible.

The firewall hostname, graph time period, and graph resolution are printed along the bottom of the graph, along with the time the graph was generated.

The firewall prints a table below the graph itself with a summarization of the data. This table contains minimums, averages, maximums, current values, in some cases 95th percentile values. In cases where units are given, hovering the mouse pointer over the unit will display a more detailed description of the unit.

Note: Totals are not displayed because the way data is stored in RRD files, accurate totals are not possible. To see total usage for traffic on network interfaces, install the **Status Traffic Totals** package.

Figure *WAN Traffic Graph* shows an example of an 8-hour graph of traffic on a firewall interface named *CABLE* with inverse enabled. The interface has a maximum utiliztion of 9.96Mbit/s during a 1 minute period.



Data Summary

	Minimum	Average	Maximum	Last	95th Percentile
inpass	16.52 kb/s	817.76 kb/s	9.96 Mb/s	0.00 b/s	3.12 Mb/s
outpass	12.44 kb/s	56.21 kb/s	262.97 kb/s	0.00 b/s	122.66 kb/s
inblock	0.00 b/s	7.27 b/s	676.90 b/s	0.00 b/s	
outblock	0.00 b/s	0.04 b/s	11.54 b/s	0.00 b/s	
inpass6	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s
outpass6	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s
inblock6	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	
outblock6	0.00 b/s	0.00 b/s	0.00 b/s	0.00 b/s	
inpass total	16.52 kb/s	817.76 kb/s	9.96 Mb/s	0.00 b/s	3.12 Mb/s
outpass total	12.44 kb/s	56.21 kb/s	262.97 kb/s	0.00 b/s	122.66 kb/s

Fig. 15: WAN Traffic Graph

Graph Settings

To change the graph, click  in the breadcrumb bar to display the graph settings panel.

Tip: The graph settings panel is hidden by default but this behavior can be changed. Navigate to **System > General Setup** and check **Monitoring Settings** to always display the settings panel by default.

The options on the settings panel are:

Left Axis / Right Axis The options here control the data dispayed on each axis. By default only the **Left Axis** is populated with a value, but both may be utilized to compare areas. First pick a **Category** (or *None*), then pick a **Graph** inside that category. The list of available categories and graphs will vary depending on the firewall configuration.

Category The general area of the desired graph: System, Traffic, Packets, Quality, Captive Portal, NTP, Queues, QueueDrops, DHCP, Cellular, Wireless, and VPN Users. These are covered in more detail later in this section.

Graph The specific graph to display from the chosen category.

Options This section of the settings panel controls how the graph itself looks, including the time span and style.

Time Period The length of time to show on the graph. The default ranges cover from 1 hour up to 4 years, or a *Custom* period may be chosen. Selecting *Custom* displays the **Custom Period** controls. All of the periods are displayed even if there is no data in a graph database going back that far. The graph will be empty for times when graphing was not active.

Resolution The smallest slice of time for which data is available on this graph. Over time, data is consolodated over longer periods so resolution is lost. For example, on a 1-hour graph it is possible to see data from one minute intervals, but on a graph including older data, it is not possible to show data that accurately since it has been averaged out. Depending on the time period of the graph it may contain *1 Minute*, *5 Minute*, *1 Hour*, or *1 Day* averages for data. Resolutions which are not possible for the given time period cannot be selected.

Inverse Used on graphs such as the traffic graph, to separate incoming and outgoing data. For example, with **Inverse** set to *On*, outbound data is represented as a negative value to more easily differentiate it from inbound data.

Custom Period When **Time Period** is set to *Custom*, the GUI displays this section to configure the custom time period for the graph.

Start Date The start date for the graph. Clicking in the field will show a calendar date picking control. Only today, or days in the past, may be selected.

Start Hour The hour of the day to start the graph using 24-hour style (`0-23`).

End Date The end date for the graph.

End Hour The end hour for the graph, exclusive. The chosen hour is not included in the graph. For example, on a graph starting at hour `10` to hour `12`, the graph covers 10:00am to 12:00pm.

Settings Click  **Show Advanced** to display additional advanced controls not typically required for average use.

Export as CSV Click this button to download the data from the graph as a `.csv` (Comma Separated Values) spreadsheet file, which can then be imported into another program for analysis.

Save as Defaults Click this button to store the current graph settings as the default configuration so this specific graph will be displayed by default on future visits to this page.

Disable/Enable Graphing This toggle will disable or enable the collection of graph data. Graphing is enabled by default. Normally this would only be disabled for diagnostic purposes or if all required graphing is handled externally.

Reset Graphing Data Clicking this button will erase all graph database files and create new, empty files.

Click  **Update Graphs** to change the graph to the selected view.

Graph Category List

There are a several different categories of graph data that the firewall can plot. Each category is covered here, but not all categories will be visible on every firewall. Some graphs must be enabled separately or will only be present if a specific feature or piece of hardware is enabled.

System Graphs

The graphs under the *System* category show a general overview of the system utilization, including CPU usage, memory usage, and firewall states.

Mbuf Clusters

The **Mbuf Clusters** graph plots the network memory buffer cluster usage of the firewall. Firewalls with many interfaces, or many CPU cores and NICs that use one interface queue per core, can consume a large number of network memory buffers. In most cases, this usage will be fairly flat, but depending on various circumstances, such as unusually high load, the values may increase. If the usage approaches the configured maximum, increase the number of buffers.

See also:

Refer to *Hardware Tuning and Troubleshooting* for information on how to increase the amount of mbufs available to the OS.

The **Mbuf Clusters** graph contains the following data sources:

Current The current number of consumed mbuf clusters

Cache The number of cached mbuf clusters

Total The total of Current and Cache

Max The maximum allowed number of mbuf clusters

Memory Graph

The **Memory** graph shows the system RAM usage broken down using the following data sources:

Active The amount of active (in use) memory

Inactive The amount of inactive memory, which was in use, but could be reallocated.

Free The amount of free memory, which is not used at all.

Cache The amount of memory used for caching by the operating system.

Wire The amount of wired memory, typically kernel memory

Note: The OS will attempt to use RAM as much as posssible for caching rather than allowing it to sit idle, so the amount of free RAM will often appear lower than expected. If memory demand increases, cached memory will be made available for use.

Processor Graph

The processor graph shows CPU usage for the firewall using the following data sources:

User Utilization The amount of processor time consumed by user processes.

Nice Utilization The amount of processor time consumed by processes with a high priority.

System Utilization The amount of processor time consumed by the operating system and kernel.

Interrupts The amount of processor time consumed by interrupt handling, which is processing hardware input and output, including network interfaces.

Processes The number of running processes.

States Graph

The states graph shows the number of system states but also breaks down the value in several ways.

State Changes The number of state changes per second, or "churn". A high value from this source would indicate a rapid number of new or expiring connections.

Filter States The total number of state entries in the states table.

Source Addresses The number of active unique source IP addresses.

Destination Addresses The number of active unique destination IP addresses.

Traffic Graphs

Traffic graphs shows the amount of bandwidth used on each available interface in *bits per second* notation. The **Graph** list contains entries for each assigned interface, as well as IPsec and individual OpenVPN clients and servers.

The traffic graph is broken down into several data sources. Aside from the total, each has an IPv4 and IPv6 equivalent. The IPv6 data sources have `6` appended to the name.

inpass The rate of traffic entering this interface that was *passed* into the firewall.

outpass The rate of traffic leaving from this interface that was *passed* out of the firewall.

inblock The rate of traffic attempting to reach this interface that was *blocked* from entering the firewall.

outblock The rate of traffic attempting to leave this interface that was *blocked* from leaving the fiewall.

inpass total The total rate of traffic (IPv4 and IPv6) that was passed inbound.

outpass total The total rate of traffic (IPv4 and IPv6) that was passed outbound.

Note: The terms "inbound" and "outbound" on these graphs are from the perspective of the firewall itself. On an external interface such as a WAN, "inbound" traffic is traffic arriving at the firewall from the Internet and "outbound" traffic is traffic leaving the firewall going to a destination on the Internet. For an internal interface, such as LAN, "inbound" traffic is traffic arriving at the firewall from a host on the LAN, likely destined for a location on the Internet and "outbound" traffic is traffic leaving the firewall going to a host on the LAN.

Packet Graphs

The packet graphs work much like the traffic graphs and have the same names for the data sources, except instead of reporting based on bandwidth used, it reports the number of *packets per second* (pps) passed. The **Graph** list contains entries for each assigned interface, as well as IPsec and individual OpenVPN clients and servers.

Packets Per Second (pps) is a better metric for judging hardware performance than Traffic throguhput as it more accurately reflects how well the hardware handles packets of any size. A circuit may be sold on a certain level of bandwidth, but hardware is more likely to be bottlenecked by an inability to handle a large volume of small packets. In situations where the hardware is the limiting factor, the **Packets** graph may show a high plateau or spikes while the traffic graph shows usage under the rated speed of the line.

Quality Graphs

The **Quality** category contains **Graph** entries that track the quality of WAN or WAN-like interfaces such as interfaces with a gateway specified or those using DHCP or PPPoE. The firewall contains one **Graph** entry per gateway, including gateways that were configured previously, but no longer exist. Graph data files for old gateways are not automatically removed so that historical data is available for future reference.

The following data sources are used to track gateway reliability:

Packet Loss The percentage of attempted pings to the monitor IP address that were lost. Loss on the graph indicates connectivity issues or times of excessive bandwidth use where pings were dropped.

Delay Average The average delay (Round-trip time, RTT) on pings sent to the monitor IP address. A high RTT means that traffic is taking a long time to make the round trip from the firewall to the monitor IP address and back. A high RTT could be from a problem on the circuit or from high utilization.

Delay Standard Deviation The standard deviation on the RTT values. The standard deviation gives an impression of the variability of the RTT during a given calculation period. A low standard deviation indicates that the connection is relatively stable. A high standard deviation means that the RTT is fluctuating up and down over a large range of values, which could mean that the connection is unstable or very busy.

Captive Portal

The **Captive Portal** category contains **Graph** entries for each Captive Portal zone, past and present. Graph data files for old zones are not automatically removed.

Concurrent The *Concurrent* graph choice shows how many users are logged in at a given point in time. As users log out or their sessions expire, this count will go down. A large number of concurrent users will not necessarily cause a strain on the portal, but it can be useful for judging overall capacity and bandwidth needs.

Logged In The *Logged In* graph shows the number of login events that occur during each polling interval. This is useful for judging how busy the captive portal daemon is at a given point in time. A large number of users logging in around the same time will put more stress on the portal daemon compared to logins that are spread out over the course of a day.

NTP

The **NTP** graph displays statistics about the NTP service and clock quality. This graph is disabled by default because it is not relevant for most use cases. The graph can be enabled at **Services > NTP**. On that page, check **Enable RRD Graphs of NTP statistics**.

See also:

For more information about these values, see the NTP Configuration Manual, NTP Query Manual, and the NTPv4 Specification.

Offset Combined clock difference between from server relative to this host.

System Jitter (sjit) Combined system jitter, which is an estimate of the error in determining the offset.

Clock Jitter (cjit) Jitter computed by the clock discipline module.

Clock Wander (wander) Clock frequency stability expressed in parts per million (PPM)

Frequency Offset (freq) Offset relative to hardware clock (In PPM)

Root Dispersion (disp) Total difference between the local clock and the primary reference clock across the network.

Queue/Queuedrops Graphs

The queue graphs are a composite of each traffic shaper queue. Each individual queue is shown, represented by a unique color.

The **Queues** category shows individual queue usage in bytes.

The **QueueDrops** category shows a count of packet drops from each queue.

DHCP

The **DHCP** category contains a graph for each interface with a DHCP server enabled. The data sources shown for DHCP are:

Leases The number of leases in use out of the configured DHCP range for the interface.

Static Leases The number of static mapping leases configured for the interface.

DHCP Range The total size of the DHCP pool available for use on the interface.

If the **Leases** count approaches the **Range** value, then a larger pool may be required for the interface. Static mappings exist outside the range, so they do not factor into the amount of leases consumed in the pool.

Cellular

On select 3G/4G devices, the firewall is able to collect signal strength data for the **Cellular** graph. The signal strength is the only value plotted on the graph.

Wireless

The **Wireless** category is present on systems containing an 802.11 wireless network device that is enabled and in-use as a client (Infrastructure, BSS mode). The following data sources are collected and displayed when acting as a wireless client:

SNR The signal-to-noise ratio for the AP the client is connected to.

Channel The wireless channel number used to reach the AP.

Rate The wireless data rate to the AP.

VPN Users

The **VPN Users** category shows the number of OpenVPN users logged in concurrently for each individual OpenVPN server.

27.2.2 Traffic Graphs

Real time traffic graphs drawn with JavaScript using NVD3 are available which update continually. These graphs can be viewed at **Status > Traffic Graph**, and an example of the graph can be found in Figure *Example LAN Graph*.

These traffic graphs show interface traffic as it happens, and give a clear view of what is happening "now" rather than relying on averaged data from the RRD graphs which are better for long-term views.

Only one interface is visible at a time, and this interface can be changed using the **Interface** drop-down list. Once an interface is chosen, the page will automatically refresh and start displaying the new graph.

Similar style traffic graphs can also be viewed on the Dashboard by adding the **Traffic Graphs** widget. Using the widget, multiple traffic graphs can be displayed simultaneously.

See also:

For more about the Dashboard, see *Dashboard*.

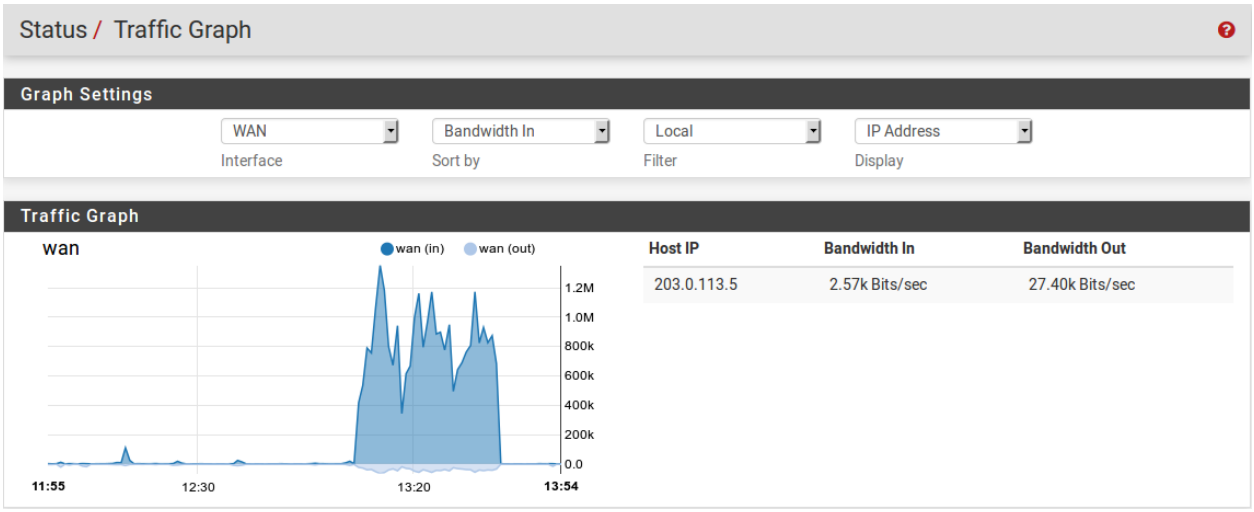


Fig. 16: Example LAN Graph

A table containing momentary glimpses of data being transferring from specific IP addresses is also displayed next to the traffic graph. These are limited to only displaying briefly, so ongoing transfers are more likely to show up than quick connections. Also, only connection from within that interface's primary subnet will be shown.

The display of the graph and table can be controlled using the following options:

Interface The firewall interface to use as the traffic source for the graph and the table.

Sort By Selects the sort order of the graph, either *Bandwidth In* or *Bandwidth Out*.

Filter Selects which type of hosts to display in the table

Local Shows only IP addresses within the interface network

Remote Shows only IP addresses that are not within the interface network

All Shows all IP addresses, inside and outside the interface network

Display Controls the display of the **Host IP** column using one of the following choices:

IP Address The IP address of the host.

Host Name The short hostname that corresponds to the IP address, as listed in DHCP static mappings, DNS Resolver host overrides, or DNS Forwarder host overrides.

Description The description that corresponds to the IP address, as listed in DHCP static mappings, DNS Resolver host overrides, or DNS Forwarder host overrides.

FQDN The fully qualified domain name that corresponds to the IP address, as listed in DHCP static mappings, DNS Resolver host overrides, or DNS Forwarder host overrides.

27.2.3 Monitoring Bandwidth Usage

With pfSense® software, there are several methods for monitoring bandwidth usage, with different levels of granularity.

pftop

If a connection is currently active, connect to the pfSense router's console (physical access or ssh) and watch the traffic flow with pftop (Option `9`).

The output can be changed to show several views (press `0-8` or `v` to cycle) and may be sorted in various ways. Press `?` for a list of available command keys while running pftop.

iftop

Install iftop from the *Package List*, then tun it from the shell (console or SSH) as follows:

```
iftop -nNpPi em0
```

Change `em0` to be the interface that should be monitored.

In the above example, `-nNpP` tells iftop to not resolve hostnames (`n`) or port numbers (`N`), and to run in promiscuous mode (`p`) and also display ports in the output (`P`).

Press `t` to cycle through various views.

trafshow

Another option for viewing real time throughput is trafshow. To install it, follow the example at *Installing FreeBSD Packages* using *trafshow* as the package name.

Once installed, run it at an SSH command prompt, run:

```
trafshow
```

Then select the interface.

Built-in Graphs

If overall per-interface usage is all that is required, there are built-in RRD graphs in pfSense software, which can be found under **Status > Monitoring**.

BandwidthD

If more detail is required, such as by client IP on the LAN interface, there is a package for bandwidthd that can be installed under **System > Packages**. Once installed, it appears under **Diagnostics > BandwidthD**.

Darkstat

Darkstat is also available in **System > Packages**. Once installed, it appears under **Diagnostics > darkstat**. It also offers bandwidth graphs for an interface, as well as traffic to/from specific IP addresses.

ntopng

If even more detail is required, the ntopng package, which can also be found under **System > Packages**, can help. It can break down detail by IP, protocol, and so on. Once installed, it appears under **Diagnostics > ntopng**. It will even track where connections were made by local PCs, and how much bandwidth was used on individual connections.

The older ntop package has been replaced by ntopng.

Due to the disk resource requirements of ntop and ntopng, it is not recommended for systems that have low CPU or RAM.

Monitoring on Multiple Interfaces

Currently, **darkstat** and **bandwidthd** do not listen on multiple interfaces. **ntopng** will listen on multiple interfaces.

Netflow

Netflow is another option for bandwidth usage analysis. Netflow is a standard means of traffic accounting supported by many routers and firewalls. Netflow collector running on a host inside the network is required to collect the data. pfSense software can export Netflow data to the collector using the *softflowd* package or the pfflowd package.

Traffic Totals

Traffic Totals is another bandwidth monitoring tool available to install as a package. See *Traffic Totals* for more information

See also:

- *Exporting NetFlow with softflowd*

27.3 Logs

Logs in pfSense software contain recent events and messages from daemons. These messages can be stored locally on a limited basis, or forwarded to a central logging server for long-term storage, better reporting, alerting, and so on.

27.3.1 System Logs

pfSense® software logs a lot of data by default, but does so in a manner that will not overflow the storage on the firewall. The logs can be viewed in the GUI under **Status > System Logs** and under `/var/log/` on the file system.

Some components such as DHCP and IPsec generate enough logs that they have their own logging tabs to reduce clutter in the main system log and to ease troubleshooting for these individual services. To view other logs, click the tab for the subsystem to view. Certain areas, such as **System**, and **VPN**, have sub-tabs with additional related options.

pfSense logs are contained in a binary circular log format called **clog**. These files are a fixed size and never grow. As a consequence of this, the log will only hold a certain amount of entries and the old entries are continually pushed out of the log as new entries are added. If log retention is an issue for an organization, the logs can be copied to another

server with syslog where they may be permanently retained or rotated with less frequency. See *Remote Logging with Syslog* later in this chapter for information about syslog.

On normal installations where logs are kept on disk, they are retained across reboots. When `/var` is in a RAM disk, the system attempts to backup the logs at shutdown and restore them when booting. If the system does not shut down cleanly, the logs will reset.

Viewing System Logs

The system logs can be found under **Status > System Logs**, on the **System** tab. This will include log entries generated by the host itself in addition to those created by services and packages which do not have their logs redirected to other tabs/log files.

As shown by the example entries in Figure *Example System Log Entries*, there are log entries from several different areas in the main system log. Many other subsystems will log here, but most will not overload the logs at any one time. Typically if a service has many log entries it will be moved to its own tab and log file.

Aug 5 18:15:57	avahi-daemon[38307]: Found user 'avahi' (UID 1003) and group 'avahi' (GID 1003).
Aug 5 18:15:41	avahi-daemon[44110]: Leaving mDNS multicast group on interface em0.IPv4 with address 192.168.10.1.
Aug 5 18:15:41	avahi-daemon[44110]: Leaving mDNS multicast group on interface tun0.IPv4 with address 192.168.100.2.
Aug 5 18:15:41	avahi-daemon[44110]: Got SIGTERM, quitting.
Aug 5 18:15:32	sshd[38258]: Accepted password for admin from 192.168.10.10 port 64864 ssh2
Aug 5 01:01:02	php: : phpDynDNS: No Change In My IP Address and/or 25 Days Has Not Past. Not Updating Dynamic DNS Entry.
Aug 5 01:01:02	php: : DynDns: Cached IP: 72.69.194.6
Aug 5 01:01:02	php: : DynDns: Current WAN IP: 72.69.194.6
Aug 5 01:01:02	php: : DynDns: _detectChange() starting.
Aug 5 01:01:02	php: : DynDns: updatedns() starting
Aug 5 01:01:02	php: : DynDns: Running updatedns()

Fig. 17: Example System Log Entries

Filtering Log Entries

Every log can be searched and filtered to find entries matching a specified pattern. This is very useful for tracking down log messages from a specific service or log entries containing a specific username, IP address, and so on.

To search for log entries:

- Navigate to **Status > System Logs** and then the tab for the log to search
- Click  in the breadcrumb bar to open the **Advanced Log Filter** panel
- Enter the search criteria, for example, place some text or a regular expression in the **Message** field
- Click  **Apply Filter**

The filtering fields vary by log tab, but may include:

Message The body of the log message itself. A word or phrase may be entered to match exactly, or use Regular Expressions to match complex patterns.

Time The timestamp of the log message. Uses month names abbreviated to three letters.

Process The *name* of the process or daemon generating the log messages, such as `sshd` or `check_reload_status`.

PID The process ID number of a running command or daemon. In cases where there are multiple copies of a daemon running, such as `openvpn`, use this field to isolate messages from a single instance.

Quantity The number of matches to return in filter results. Setting this value higher than the number of log entries in the log file will have no effect, but setting it higher than the current display value will temporarily show more log messages.

The **Firewall** log tab has a different set of filtering fields:

Source IP Address The source IP address listed in the log entry.

Destination IP Address The destination IP address listed in the log entry.

Pass Check this option to only match log entries that passed traffic.

Block Check this option to only match log entries that blocked traffic.

Interface The friendly description name of the interface to match (e.g. `WAN`, `LAN`, `OPT2`, `DMZ`)

Source Port The source port of the log entry to match, if the protocol uses ports.

Destination Port The destination port of the log entry to match, if the protocol uses ports.

Protocol The protocol to match, such as TCP, UDP, or ICMP.

Protocol Flags For TCP, this field matches the TCP flags on the log entry, such as `SA` (SYN+ACK) or `FA` (FIN+ACK)

The filter pane is hidden by default but it can be included on the page at all times by checking **Log Filter** under **System > General Setup**.

See also:

- *Log Settings*
- *Remote Logging with Syslog*
- *Working with Log Files*
- *Adjusting the Size of Log Files*

27.3.2 Log Settings

Log settings on pfSense® software may be adjusted in two different ways:

- Globally at **Status > System Logs** on the **Settings** tab
- On each log tab where settings can override the global defaults

To change these settings click  in the breadcrumb bar while viewing a log.

Each of these methods will be explained in detail in this section.

The global options area contains more options than the per-log settings. Only differences will be covered in detail for the per-log settings.

Global Log Settings

The global log options under **Status > System Logs** on the **Settings** tab include:

In the GUI, the **Settings** tab under **Status > System Logs** controls how the logging system behaves.

Log Message Format The format of messages logged by the system log daemon (`syslogd`) for local and remote logs. Both formats are handled the same way locally, but remote syslog servers may prefer one format or the other. Check the documentation of the syslog server for details.

BSD (RFC 3164, default) The default log format used by previous versions of pfSense software and natively used by FreeBSD.

syslog (RFC 5424, with RFC 3339 microsecond-precision timestamps A modern syslog message format with more precise timestamps. Also includes the hostname.

Forward/Reverse Display By default the logs are displayed in their natural order with the oldest entries at the top and the newest entries at the bottom. Some administrators prefer to see the newest entries at the top, which can be accomplished by checking this box to flip the order.

GUI Log Entries The number of log entries to display in the log tabs of the GUI by default. This does not limit the number of entries in the file, only what is shown on the page at the time. The default value is `50`. The actual log files may contain much more than the number of lines to display, depending on the **Log File Size**.

Log File Size (Bytes) (<= 2.4.x) The size of the *circular log file*. The size of the file directly controls how many entries it can contain. The default log size is approximately `500,000` bytes (500KB).

There are roughly 20 log files, so any increase in file size will result in 20 times larger total disk utilization from logs. The current total log size and remaining disk space are displayed for reference. At the default size, the logs will hold about 2500 entries on average but it may be significantly more or less depending on the size of individual log entries.

> **Warning:** The new log size will **not** take effect until a log is cleared or reinitialized. This may be done individually from each log tab or it can be done for all logs using the  **Reset Log Files** button on this page. See *Adjusting the Size of Log Files* for more.

Log Packets from Default Block Rules Checked by default. When enabled, the default deny rule, which blocks traffic not matched by other rules, will log entries to the firewall log. Typically these log entries are beneficial, but in certain rare use cases they may produce undesirable log entries that are made redundant by custom block rules with logging enabled.

Log Packets from Default Pass Rules Unchecked by default. When set, logging will occur for packets matching the default pass out rules on interfaces. Setting this option will generate a large amount of log data for connections outbound from the firewall. The best practice is to only enable this for brief periods of time while performing troubleshooting or diagnostics.

Log Packets from Block Bogon Networks Rules Checked by default. When checked, if an interface has **Block Bogon Networks** active, packets matching that rule will be logged. Uncheck to disable the logging.

Log Packets from Block Private Networks Rules Checked by default. When checked, if an interface has **Block Private Networks** active, packets matching that rule will be logged. Uncheck to disable the logging.

Web Server Log When checked, log messages from the Web GUI process, `nginx`, will be placed in the main system log. On occasion, especially with Captive Portal active, these messages can be frequent

but irrelevant and clutter the log contents.

Raw Logs When checked, this setting disables log parsing, displaying the raw contents of the logs instead. The raw logs contain more detail, but they are much more difficult to read. For many logs it also stops the GUI from showing separate columns for the process and PID, leaving all of that information contained in the **Message** column.

Show Rule Descriptions Controls if, and where, the firewall log display will show descriptions for the rules that triggered entries. Displaying the rule descriptions causes extra processing overhead that can slow down the log display, especially in cases where the view is set to show a large number of entries.

Don't load descriptions When selected this choice will not display any rule descriptions. The description may still be viewed by clicking the action column icon in the firewall log view (e.g.  or ).

Display as column The default for new installations. Adds the rule description in a separate column. This works best if the descriptions are short, or the display is wide.

Display as second row Adds a second row to each firewall log entry containing the rule description. This choice is better for long rule descriptions or narrow displays.

Tip: If the firewall logs display slowly with rule descriptions enabled, select *Don't load descriptions* for faster performance.

Local Logging When checked, local logs are not retained. They are not written to disk nor are they kept in memory. While this saves on disk writes, it necessitates the use of remote logging so that information is not lost. This is not a best practice, as having local logs is vital for the vast majority of use cases.

Reset Log Files This button clears the data from all log files and reinitialize them as new, empty logs. This must be done after changing the log file sizes (2.4.x), and can also be used to clear out irrelevant/old information from logs if necessary.

> **Warning:** Resetting the log files will not save the other options on the page. If options on this page have been changed, click **Save** before attempting to reset the log files.

Click **Save** to store the new settings. The remaining options on this screen are discussed in *Remote Logging with Syslog*.

Log Rotation Settings

Starting with pfSense software version 2.5.0, the system logs are kept in a plain text format and periodically rotated. The options in this section control how the system handles log rotation.

Note: The options in this section of the page are global only, and cannot be changed for individual logs.

Log Rotation Size (Bytes) This field controls the size at which the system will rotate logs. The default size is `500` KiB per log file. There are nearly 20 log files, so plan space accordingly.

This **does not** account for space used by rotated log files.

Note: Increasing this value allows every log file to grow to the specified size, so disk usage may increase significantly. Log file sizes are checked once per minute to determine if rotation is necessary, so a rapidly growing log file may exceed this value.

Log Compression The type of compression to use when the system rotates log files. Compressing rotated log files saves disk space, and the compressed logs remain available for display and searching in the GUI. Though processing large compressed files can be time consuming, most use cases will not notice significant slowness.

The types of compression available are `bzip2` (default), `gzip`, `xz`, `zstd`, and `none` (disables compression). All of the options which use compression are reasonably fast and offer good compression rates. Some may compress better than others, others are slightly faster, but ultimately the decision is up to the environment and the administrators.

Warning: The type of compression used by all log files must be identical. When changing this value, the system must remove all previously rotated compressed log files.

On certain systems, disabling compression (set to `none`) is the best course of action. Examples include:

- Systems using large log file sizes, which may take too long to compress
- Slower systems which may take too long to compress or search the log files even at default sizes
- Systems using ZFS which by default will already compress disk contents

Log Retention Count The number of rotated log files to keep before the oldest copy is removed. Keeping more log files will consume more disk space, but compressed logs files do not consume nearly as much space as decompressed logs.

Per-Log Settings

To change per-log settings, visit the log tab to change and then click  in the breadcrumb bar to expand the settings panel.

On this panel, several options are displayed. Most of the options will show the global default value or have a **General Logging Options Settings** choice which will use the global value and not the per-log value.

The per-log settings panel for each tab only displays options relevant to that log. For example, the options to log default block or pass rules are displayed only when viewing the **Firewall** log tab.

Each per-log settings panel has at least the following options: **Forward/Reverse Display**, **GUI Log Entries**, **Log File Size (Bytes)** (2.4.x), and **Formatted/Raw Display**. For each of these, a value which will only apply to this log may be set. For more information on how these options work, see *Global Log Settings* above.

Click **Save** to store the new log settings.

Note: If the log file size was changed, after saving, open the settings panel again and click the  **Clear Log** button to reset the log using the new size.

See also:

- *Remote Logging with Syslog*
- *Accessing Firewall Services over IPsec VPNs*
- *Working with Log Files*
- *Adjusting the Size of Log Files*

27.3.3 Remote Logging with Syslog

The **Remote Logging** options under **Status > System Logs** on the **Settings** tab allow syslog to copy log entries to a remote server.

The logs kept by pfSense® software on the firewall itself are of a finite size. Copying these entries to a syslog server can aid troubleshooting and allow for long-term monitoring. Having a remote copy can also help diagnose events that occur before a firewall restarts or after they would have otherwise been lost due to clearing of the logs or when older entries are cycled out of the log, and in cases when local storage has failed but the network remains active.

Warning: Corporate or local legislative policies may dictate the length of time logs must be retained from firewalls and similar devices. If an organization requires long-term log retention for their own or government purposes, a remote syslog server is required to receive and retain these logs.

Warning: Logs sent using this method are delivered in the clear (not encrypted) unless the logs are sent through a VPN or using a mechanism such as *Stunnel package*. As an alternative, consider using the **syslog-ng** package which supports encrypted syslog.

The following options are available for remote logging:

Source Address Controls where the `syslog` daemon binds for sending out messages. In most cases, the default (*Any*) is the best option, so the firewall will use the address nearest the target. If the destination server is across a tunnel mode IPsec VPN, however, choosing an interface or Virtual IP address inside the local Phase 2 network will allow the log messages to flow properly over a tunnel.

IP Protocol When choosing an interface for the **Source Address**, this option gives the `syslog` daemon a preference for either using IPv4 or IPv6, depending on which is available. If there is no matching address for the selected type, the other type is used instead.

Remote Log Servers Enter up to three remote servers using the boxes contained in this section. Each remote server can use either an IP address or hostname, and an optional UDP port number. If the port is not specified, the default *syslogd* port, `514`, is assumed.

A syslog server is typically a server that is directly reachable from the firewall on a local interface. Logging can also be sent to a server across a VPN.

Warning: Do not send log data directly across any WAN connection or unencrypted site-to-site link, as it is plain text and could contain sensitive information.

Note: The syslog daemon only supports sending messages over UDP. To send syslog messages over TCP, consider using the **syslog-ng** package.

Remote Syslog Contents The options in this section control which log messages will be sent to the remote log server.

Everything When set, all log messages from all areas are sent to the server.

System Events Main system log messages that do not fall into other categories.

Firewall Events Firewall log messages in raw format. The format of the raw log is covered in *Raw Filter Log Format*.

DNS Events Messages from the DNS Resolver (`unbound`), DNS Forwarder (`dnsmasq`), and from the `filterdns` daemon which periodically resolves hostnames in aliases.

DHCP Events Messages from the IPv4 and IPv6 DHCP daemons, relay agents, and clients.

PPP Events Messages from PPP WAN clients (PPPoE, L2TP, PPTP)

General Authentication Events Log messages about authentication events, such as for the GUI or certain types of VPNs.

Captive Portal Events Messages from the Captive Portal system, typically authentication messages and errors.

VPN Events Messages from VPN daemons such as IPsec and OpenVPN, as well as the L2TP server and PPPoE server.

Gateway Monitor Events Messages from the gateway monitoring daemon, `dpinger`

Routing Daemon Events Routing-related messages such as UPnP/NAT-PMP, IPv6 routing advertisements, and routing daemons from packages like OSPF, BGP, and RIP.

Network Time Protocol Events Messages from the NTP daemon and client.

Wireless Events Messages from the Wireless AP daemon, `hostapd`.

To start logging remotely:

- Navigate to **Status > System Logs** on the **Settings** tab
- Check **Send log messages to remote syslog server**
- Configure the options as described above
- Click Save to store the changes.

If a syslog server is not already available, it is fairly easy to set one up. Almost any UNIX or UNIX-like system can be used as a syslog server. FreeBSD is described in the following section, but others may be similar.

Setup Syslog on the Logging Host

FreeBSD

First, configure the syslog server to accept remote connections which means running it with the `-a <subnet>` or similar flag.

On FreeBSD, edit `/etc/rc.conf` and add this line:

```
syslogd_flags=" -a 192.168.1.1 "
```

Where `192.168.1.1` is the IP address of the pfSense firewall.

More complex allow rules for syslog are also possible, like so:

```
syslogd_flags=" -a 10.0.10.0/24:*"
```

Using that parameter, syslog will accept from any IP address in the 10.0.10.0 subnet (mask 255.255.255.0) and the messages may come from any UDP port.

Now, edit `/etc/syslog.conf` and add a block at the bottom:

```
!*
+*

+pfsense
*.*                /var/log/pfsense.log
```

Where `pfSense` is the hostname of the pfSense firewall. An entry may also need to be added in **/etc/hosts** for that system, depending on the DNS setup. Logs may be split separate files. Use the **/etc/syslog.conf** file on the pfSense firewall for more details on which logging facilities are used for specific items.

```
192.168.1.1            pfsense     pfsense.example.com
```

The log file may also need to be created manually with proper permissions:

```
touch /var/log/pfsense.log
chmod 640 /var/log/pfsense.log
```

Now restart syslog:

```
/etc/rc.d/syslogd restart
```

Windows

Setting this up on Windows entirely depends on which syslog server is being used. Consult the documentation for more information on configuration.

There is a free multi-purpose utility that can act as a syslog server, which can be found here: http://tftpd32.jounin.net/

Kiwi Syslog Server is free for up to 5 devices. http://www.kiwisyslog.com/downloads.aspx

Linux

Configuration of the system logger on Linux depends on the distribution. Consult the distribution's documentation on how to change the behavior of **syslogd**. It should be similar in many cases to the alterations in the FreeBSD section.

OpenBSD

The configuration for OpenBSD is similar to FreeBSD, with the following notes:

1. The option to accept remote syslog events is `-u`.
2. This option may be enabled using *rcctl(8)*:

```
rcctl set syslogd flags -u
```

1. To restart the syslogd service:

```
rcctl restart syslogd
```

Other Logging Servers

Other log systems such as Splunk, ELSA, or ELK may also be used but the methods for implementing them are beyond the scope of this document. If such a system is syslog-compatible, then the pfSense software side should be fairly simple to setup as it would be for any other syslog system.

27.3.4 Adjusting the Size of Log Files

pfSense® software stores its logs in *binary circular log files* that never grow in size. These are known as "clog" files. The fixed size prevents the logs from filling up available storage space and eliminates the need for rotation, but the down side is that the logs wrap around once full, losing older messages in the process. These log files are held in `/var/log` which may optionally be a RAM disk.

The **Log File Size (Bytes)** field in the *Log Settings* at **Status > System Logs**, **Settings** tab, allows the user to specify the amount of storage to allocate per log file.

Short Version

To change the log file sizes:

- Navigate to **Status > System Logs**, **Settings** tab
- Enter a new value in **Log File Size (Bytes)**, being careful not to overfill the disk containing the logs.
- Click **Save**
- Click **Reset Log Files**

Details

The default size for a log file is *511488* (~500KB) which can generally hold between 2000-3000 log entries but varies by entry size. The time span covered by logs depends entirely on how much data is logged. A quiet log file could contain months or even years of information, a busy log file may only contain minutes.

Underneath the text for **Log File Size (Bytes)** the current and available disk space is displayed based on the current log file sizes and their location. For example:

```
Disk space currently used by log files: 9.8M. Remaining disk space for log files: 11G
```

There are approximately 20 log files affected by the size control. The value entered in **Log File Size (Bytes)** is for a single log file, so the actual usage will be approximately **20 times** that value. As shown above, with the default value of 511488 (~500KB), the firewall uses nearly 10MB of total log space. If the size of the logs is increased to *1024000* (1MB), then nearly 20MB would be used for logs. Be certain before changing the **Log File Size (Bytes)** value that the disk has enough space to hold all of the log files.

The change to the log size only takes effect when log files are reset. Which only happens when the logs are manually reset. Log files may be reset individually using the **Clear** button on the various log tabs, or by using the master **Reset Log Files** button on the *Log Settings* page.

27.3.5 Working with Log Files

pfSense® software version 2.5.0 uses plain text log files which can be used by a variety of traditional shell utilities.

The firewall periodically rotates log files to keep their size in check. The rotation behavior is controlled by the log settings (*Log Rotation Settings*). There is one main log file, plus a number of rotated log files. The rotated log files are compressed by default. The GUI understands each compression option and will display and search contents of rotated log files in addition to the main log file. This adds processing time but vastly increases the amount of log data available to the GUI.

pfSense® software versions older than 2.5.0 use a binary circular log format known as `clog` to maintain a constant log size without the need for rotation. As `syslogd` writes new entries to a `clog` file, it removes older entries automatically. As such, the older data is lost.

Though there were multiple benefits to binary circular logs, such as restricting log file sizes, the downsides were too significant on modern systems. Among other reasons, binary circular logs were not very flexible, could not be used directly by shell utilities, were susceptible to corruption, and could not reliably store larger amounts of log data. Furthermore, the original justification for size restrictions were primarily based on hardware choices from over a decade ago. Hardware, even embedded system hardware, is much more capable now.

Viewing Log Contents (2.5.0 and later)

To view the contents of a log, use common shell utilities, such as `cat`, `grep`, and so on:

```
cat /var/log/filter.log
grep -i "error" /var/log/system.log
```

To follow the contents of a log file in real time, use `tail -f` or `tail -F`. The latter form will also follow the log to a new file after rotation.

```
tail -F /var/log/filter.log
```

In addition to the main log file, the rotated log files can be viewed and searched by passing them through utilities specific to the format with which they are compressed. For example, the default compression type is `bzip2`, so use `bzcat`, or `bzgrep`:

```
bzcat /var/log/filter.log.0.bz2
bzgrep -i "error" /var/log/system.log.0.bz2
```

Additional utilities can be utilized by piping the output.

The following list contains the different compression options and a sample of utilities which can parse their contents:

bzip2 (`*.log.<number>.bz2`) `bzcat`, `bzgrep`, `bzless`.

gzip (`*.log.<number>.gz`) `zcat`, `zgrep`, `zless`.

xz (`*.log.<number>.xz`) `xzcat`, `xzgrep`, `xzless`.

zstd (`*.log.<number>.zst`) `zstdcat`, `zstdgrep`, `zstdless`.

none (`*.log.<number>`) `cat`, `grep`, `less`, plus anything else capable of parsing text files.

Viewing Log Contents (< 2.5.0, clog)

On versions of pfSense software before 2.5.0, the *clog* command must be used to view the contents of binary circular log files from the shell:

```
clog /var/log/filter.log
```

The output of that command may then be piped to tools like `grep`:

```
clog /var/log/system.log | grep -i "error"
```

To follow the log files in a manner like `tail -f`, use `clog -f`:

```
clog -f /var/log/filter.log
```

The command prints the entire contents of the log file to the console, and then prints new entries as they are written.

Log Sizes

The default is 500 KiB per log file, and there are around 20 log files. When increasing log sizes, keep disk space in mind. There is a disk space indicator for the filesystem containing the logs under the **Log File Size (Bytes)** text description on *Log Settings*.

On 2.5.0, space for rotated log files is in addition to this limit. The rotation settings are described in *Log Rotation Settings*.

The log files created for use by pfSense with *clog* are a fixed size that holds a certain amount of *data* total, not log entries. As such, the number of log entries may vary widely depending on the length of the lines and message content. Log files typically contain between 2000 and 4000 entries, but it could be much more or less than that.

The GUI only shows 50 lines per log by default but the files contain many more entries. See *Log Settings* for more information on that setting.

27.3.6 Viewing the Firewall Log

The firewall creates log entries for each rule configured to log and for the default deny rule. There are several ways to view these log entries, each with varying levels of detail. There is no clear "best" method since it depends on the preferences and skill level of the firewall administrators, though using the GUI is the easiest method.

Tip: The logging behavior of the default deny rules and other internal rules can be controlled using the **Settings** tab under **Status > System Logs**. See *Log Settings* for details.

Like other logs, the firewall log only retains a certain number of entries. If the needs of an organization require a permanent record of firewall logs for a longer period of time, see *Remote Logging with Syslog* for information on copying these log entries to a syslog server as they happen.

See also:

- *Troubleshooting Blocked Log Entries for Legitimate Connection Packets*
- *Working with Log Files*
- *Log Settings*
- *Filtering Log Entries*

Viewing in the WebGUI

The firewall logs are visible in the WebGUI at **Status > System Logs**, on the **Firewall** tab. From there, the logs can be viewed as a parsed log, which is easier to read, or as a raw log, which contains more detail. There is also a setting to show these entries in forward or reverse order. If the order the log entries being displayed is unknown, check the timestamp of the first and last lines, or check *Log Settings* for information on how to view and change these settings.

The parsed WebGUI logs, seen in Figure *Example Log Entries Viewed From The WebGUI*, are in 6 columns: **Action**, **Time**, **Interface**, **Source**, **Destination**, and **Protocol**.

Action Shows what happened to the packet which generated the log entry (e.g. pass or block)

The **Action** icon is a link which, when clicked, looks up and displays the rule which caused the log entry. More often than not, this says "Default Deny Rule", but when troubleshooting rule issues it can help narrow down suspects.

Time The time that the packet arrived.

Interface Where the packet entered the firewall.

The GUI prints a ▶ character next to the interface if a rule matched a packet in the *outbound* direction. The vast majority of rules match in the inbound direction, so the direction is omitted in that case.

Rule The firewall rule description and ID number which generated the log entry, if available. This column only appears when rule descriptions are set to appear in a separate column. They may also be shown in a separate row, or disabled entirely. See *Log Settings* for details.

Source The source IP address and port.

The  icon next to the source and destination IP addresses, when clicked, makes the firewall perform a DNS lookup on the IP address. If the address has a valid hostname it will be displayed underneath the IP address in all instances of that address on the page.

The  icon next to the source IP address and the  icon next to the destination IP address are for adding firewall rules with EasyRule. See *Using EasyRule to Add Firewall Rules* for details.

Destination The destination IP address and port.

Protocol The protocol of the packet, e.g. ICMP, TCP, UDP, etc.

Log entries for TCP packets have extra information appended to the protocol field displaying TCP flags present in the packet. These flags indicate various connection states or packet attributes. The meanings for each flag are outlined in *TCP Flags*.

Action	Time	Interface	Source	Destination	Protocol
✖	Aug 3 08:59:02	WAN	198.51.100.1:67	198.51.100.2:68	UDP
✖	Aug 3 15:02:10	WAN	198.51.100.108:138	198.51.100.255:138	UDP
✖	Aug 3 15:02:10	WAN	198.51.100.108:138	198.51.100.255:138	UDP
✖	Aug 3 15:14:02	WAN	198.51.100.108:138	198.51.100.255:138	UDP
✖	Aug 3 15:14:02	WAN	198.51.100.108:138	198.51.100.255:138	UDP

Fig. 18: Example Log Entries Viewed From The WebGUI

The GUI can also filter log output to find specific entries, so long as they exist in the current log. Click  to display the filtering options. See *Filtering Log Entries* for more information.

Viewing from the Console Menu

Option `10` from the console menu views and follows the `filter.log` in real time. An easy example is a log entry like that seen above in Figure *Example Log Entries Viewed From The WebGUI*:

```
Aug  3 08:59:02 master filterlog: 5,16777216,,1000000103,igb1,match,block,in,4,0x10,,
↪128,0,0,
none,17,udp,328,198.51.100.1,198.51.100.2,67,68,308
```

This single line shows that the log entry was triggered by rule id `1000000103`, which resulted in a block action on the `igb1` interface. The source and destination IP addresses are shown near the end of the log entry, followed by the source and destination port. Packets from other protocols may show significantly more data.

See also:

See *Raw Filter Log Format* for details on the format of the filter log file.

Viewing from the Shell

When using the shell, either from SSH or from the console, there are numerous options available to view the filter logs.

When directly viewing the contents of the log file, the log entries can be quite complex and verbose.

For information on viewing logs from the shell, see *Working with Log Files*.

Viewing parsed log output in the shell

There is a simple log parser written in PHP which can be used from the shell to produce reduced output instead of the full raw log. To view the parsed contents of the current log, run:

2.5.0 and later:

```
# cat /var/log/filter.log | filterparser.php
```

Older versions:

```
# clog /var/log/filter.log | filterparser.php
```

The script prints the log entries one per line, with simplified output:

```
Aug  3 08:59:02 block igb1 UDP 198.51.100.1:67 198.51.100.2:68
```

Finding the rule which caused a log entry

When viewing one of the raw log formats, the log includes the rule ID number for an entry. This rule number can be used to find the rule which caused the match. The following example locates the rule with id `1000000103`:

```
# pfctl -vvsr | grep 1000000103
@5(1000000103) block drop in log inet all label "Default deny rule IPv4"
```

As shown in the above output, this was the default deny rule for IPv4.

27.3.7 Raw Filter Log Format

The raw filter log output format generated by pfSense software for its internal filter log, and the log output transmitted over syslog to remote hosts, is a single line containing comma-separated values.

Plain text layout

In general terms, here is the content of the log file. For a more technical representation with greater detail, see the next section:

```
<Timestamp> <Hostname> filterlog: <CSV data>
```

Note: Hostname is not included in syslog data sent to remote log hosts

CSV Data has many common fields and some that vary by protocol:

Common fields:

- Rule Number
- Sub rule number
- Anchor
- Tracker - unique ID per rule, tracker ID is stored with the rule in config.xml for user added rules, or check /tmp/rules.debug
- Real interface (e.g. em0)
- Reason for the log entry (e.g. match)
- Action taken that resulted in the log entry (e.g. block, pass)
- Direction of the traffic (in/out)
- IP version (4 for IPv4, 6 for IPv6)

IPv4:

- TOS
- ECN
- TTL
- ID
- Offset
- Flags
- Protocol ID
- Protocol text (tcp, udp, etc)

IPv6:

- Class
- Flow Label
- Hop Limit
- Protocol

- Protocol ID

IPv4 or IPv6:

- Length
- Source IP
- Destination IP

For TCP and UDP (Proto ID 6 or 17) on IPv4 or IPv6

- Source Port
- Destination Port
- Data Length

TCP Only:

- TCP Flags
- Sequence Number
- ACK
- Window
- URG
- Options

ICMP:

- ICMP Type, used to choose between the following possibilities

ICMP Echo Request/Reply

- ICMP ID
- ICMP Sequence

ICMP Protocol Unreachable

- ICMP Destination IP
- ICMP Protocol ID

ICMP Port Unreachable

- ICMP Destination IP
- ICMP Protocol ID
- ICMP Port Number

ICMP unreachable (other), time exceeded, parameter problem, redirect, mask reply:

- ICMP Description

ICMP Need Frag

- ICMP Destination IP
- ICMP MTU

ICMP tstamp

- ICMP ID
- ICMP Sequence

ICMP tstamp reply

- ICMP ID
- ICMP Sequence
- ICMP otime
- ICMP rtime
- ICMP ttime

ICMP default:

- ICMP Description

CARP (Protocol ID 112):

- Type
- TTL
- VHID
- Version
- Advskew
- Advbase

BNF / Grammar

For more technical purposes, this is a BNF format representation of the log output. It is not a 100% complete BNF as the exact contents of many of the fields are beyond the scope of this document as they aren't generally relevant to typical logging, but they are included in the log entries for completeness. Consult a reference on IP packet headers for more information.

```
<log-entry> ::= <time-stamp> <host-name> "filterlog:" <log-data>

<log-data> ::= <rule-number>,<sub-rule-number>,<anchor>,<tracker>,<real-interface>,
↪<reason>,<action>,<direction>,<ip-version>[,<ip-specific-data>]

<rule-number> ::= <integer> -- Rule number in the pf Ruleset
<sub-rule-number> ::= <integer> -- Sub rule number in the pf Ruleset (not typically␣
↪significant for general use)
<anchor> ::= <text> -- Anchor name in which the rule exists
<tracker> ::= <integer> -- Unique ID per rule, tracker ID is stored with the rule in␣
↪config.xml for user added rules, or check /tmp/rules.debug
<real-interface> ::= <text> -- Real interface for the log entry (e.g. em0)
<reason> ::= <text> | "unkn(%u)" -- Reason for the log entry (typically "match")
<action> ::= "pass" | "block" | "unkn(%u)" -- Action taken that resulted in the log␣
↪entry
<direction> ::= "in" | "out" | "unkn(%u)" -- Direction of the logged traffic
<ip-version> ::= "4" | "6" -- IPv4 or IPv6
<ip-specific-data> ::= (<ipv4-specific-data>|<ipv6-specific-data>),<ip-data>[,
↪<protocol-specific-data>]

<ipv4-specific-data> ::= <tos>,<ecn>,<ttl>,<id>,<offset>,<flags>,<protocol-id>,
↪<protocol-text>

<tos> ::= <empty> | <hex> -- Type of Service identification
<ecn> ::= <empty> | -- Explicit Congestion Notification
```

(continues on next page)

(continued from previous page)

```
<ttl> ::= <integer> -- Time To Live (TTL) of the packet
<id> ::= <integer> -- ID of the packet
<offset> ::= <integer> -- Fragment offset
<flags> ::= "none" | <text> -- IP Flags (NOT TCP flags -- those are later)
<protocol-id> ::= <integer> -- IP protocol ID (e.g. 6 for TCP, 17 for UDP)
<protocol-text> ::= "tcp" | "udp" | "icmp" | <text> -- IP protocol text (examples␣
↪given)
<ipv6-specific-data> ::= <class>,<flow-label>,<hop-limit>,<protocol-text>,<protocol-
↪id>

<class> ::= <hex> -- ToS traffic class
<flow-label> ::= <data> -- Flow label
<hop-limit> ::= <integer> -- Hop Limit (similar to IPv4 TTL)
<protocol-text> ::= "tcp" | "udp" | "icmp" | <text> -- IP protocol text (examples␣
↪given)
<protocol-id> ::= <integer> -- IP protocol ID (e.g. 6 for TCP, 17 for UDP)
<ip-data> ::= <length>,<source-address>,<destination-address>

<length> ::= <integer> -- Length of the packet in bytes
<source-address> ::= <ip-address> -- The source IP address of the logged traffic
<destination-address> ::= <ip-address> -- The destination IP address of the logged␣
↪traffic
<protocol-specific-data> ::= <tcp-data> | <udp-data> | <icmp-data> | <carp-data>

<tcp-data> ::= <source-port>,<destination-port>,<data-length>,<tcp-flags>,<sequence-
↪number>,<ack-number>,<tcp-window>,<urg>,<tcp-options>

<source-port> ::= <integer> -- Source port number
<destination-port> ::= <integer> -- Destination port number
<data-length> ::= <integer> -- Data/payload length
<tcp-flags> ::= [S][A][.][F][R][P][U][E][W] -- TCP Flags
<sequence-number> ::= <integer> -- TCP Sequence ID
<ack-number> ::= <integer> -- ACK number
<tcp-window> ::= <integer> -- Windows size
<urg> ::= <data> -- Urgent pointer data
<tcp-options> ::= <data> -- TCP Options
<udp-data> ::= <source-port>,<destination-port>,<data-length>

<icmp-data> ::= <icmp-type>,(<echo-data> | <unreachproto-data> | <unreachport-data> |
↪<other-unreachable-data> | <needfrag-data> | <tstamp-data> | <tstampreply-data> |
↪<icmp-default-data>)

<icmp-type> ::= <echo-type> | "unreachproto" | "unreachport" | <other-unreachable> |
↪"needfrag" | "tstamp" | "tstampreply" | <text>
<echo-type> ::= "request" | "reply"
<other-unreachable> ::= "unreach" | "timexceed" | "paramprob" | "redirect" |
↪"maskreply"
<echo-data> ::= <icmp-id>,<icmp-sequence>

<icmp-id> ::= <integer> -- ID of the echo request/reply
<icmp-sequence> ::= <integer> -- Sequence number of the echo request/reply
<unreachproto-data> ::= <icmp-destination-ip-address>,<unreachable-protocol-id>

<icmp-destination-ip-address> ::= <ip-address> -- Original destination address of the␣
↪connection that caused this notification
<unreachable-protocol-id> ::= <integer> -- Protocol ID number that was unreachable
<unreachport-data> ::= <icmp-destination-ip-address>,<unreachable-protocol-id>,
↪<unreachable-port-number>
```

(continues on next page)

(continued from previous page)

```
<unreachable-port-number> ::= <integer> -- Port number that was unreachable
<other-unreachable-data> ::= <icmp-description>

<icmp-description> ::= <text> -- Description from the ICMP packet
<needfrag-data> ::= <icmp-destination-ip-address>,<icmp-mtu>

<icmp-mtu> ::= <integer> -- MTU to use for subsequent data to this destination
<tstamp-data> ::= <icmp-id>,<icmp-sequence>

<tstampreply-data> ::= <icmp-id>,<icmp-sequence>,<icmp-otime>,<icmp-rtime>,<icmp-
↪ttime>

<icmp-otime> ::= <unix-timestamp> -- Originate Timestamp
<icmp-rtime> ::= <unix-timestamp> -- Receive Timestamp
<icmp-ttime> ::= <unix-timestamp> -- Transmit Timestamp
<icmp-default-data> ::= <icmp-description>

<carp-data> ::= <carp-type>,<carp-ttl>,<vhid>,<version>,<advbase>,<advskew>

<carp-type> ::= <text> -- Type of CARP/VRRP
<carp-ttl> ::= <integer> -- Time to Live
<vhid> ::= <integer> -- Virtual Host ID
<version> ::= <integer> -- CARP Version
<advbase> ::= <integer> -- Advertisement base timer interval (seconds)
<advskew> ::= <integer> -- Advertisement skew (1/256 of a second)
```

27.3.8 Gateway Logs

The gateway logs can be found through the pfSense® webGUI under **Status > System Logs** on the **System/Gateways** sub-tab.

This log contains entries from the gateway monitoring daemon, *dpinger*, which can generate a significant amount of logging with many gateways to monitor.

The entries found here will record events such as when a gateway is down, or in an alarm state, or has returned to an online state.

27.3.9 NTP Logs

The NTP daemon Log view will show any logs generated by the *Network Time Protocol daemon* and logs from the NTP client *ntpdate* that performs large time skew updates.

27.3.10 Package Logs

In pfSense® software, packages which support logging to this central location will have their logs displayed here. If a package's logs do not show up, contact the package maintainer to see if the package can be updated to support this mechanism.

Some packages will log to the main system log or a related tab inside the system logs (**Status > System Logs**). Others may keep their own logs in a separate location. Some packages, such as Squid and Snort, offer configuration options to control where and how logs are made. Some logs may need to be viewed outside the webGUI or via **Diagnostics > Command**.

27.3.11 PPP Logs

The **PPP** logs tab displays any events from the PPP system for WAN type connections, not locally-hosted servers. This would be for WANs that connect using PPPoE, 3G networks or in some rare cases, dialup, etc.

27.3.12 Resolver Logs

The Resolver logs are located at **Status > System Logs** on the **System/Resolver** tab.

This log contains entries from DNS-related processes. These include the DNS Forwarder (dnsmasq), Unbound/DNS Resolver, the filterdns process that monitors for updates in hostnames for Aliases/IPsec/etc, and the BIND package.

27.3.13 Routing Logs

The Routing logs are located at **Status > System Logs** on the **System/Routing** tab.

This log contains entries from routing-related processes for both IPv4 and IPv6, including:

- radvd (IPv6 Router Advertisements)
- zebra (FRR)
- ospfd (FRR)
- bgpd (FRR)
- bfdd (FRR)
- pimd (PIMD)
- igmpproxy (IGMP Proxy)
- miniupnpd (UPnP/NAT-PMP)

27.3.14 IPsec Logs

The IPsec logs show output from the IPsec daemon, handled by **strongswan**. Normal output, successful connections, as well as errors are all displayed here.

Where possible, if a log message contains an IP address of a configured IPsec tunnel, that tunnel's description is prepended to the log entry.

Entries found in these logs are covered in depth in *IPsec Logs*, and some errors are covered in *Troubleshooting IPsec VPNs*.

27.3.15 OpenVPN Logs

The OpenVPN logs found through the pfSense® webGUI at **Status > System Logs** and the **OpenVPN** tab show output from the OpenVPN daemon(s) in use, both clients and servers. Messages are shown in the logs for successful connections as well as failures and errors.

If there are no log entries for a server after the process starts, traffic likely is not reaching the OpenVPN daemon. Check the WAN-side firewall rules and the address/port used by the client.

There are several OpenVPN troubleshooting articles found in *Troubleshooting*.

27.3.16 Captive Portal Authentication Logs

The Captive Portal Authentication Logs are available through the pfSense® webGUI at **Status > System Logs**, on the **Portal Auth** tab. The logs list login information from the **Captive Portal** system. Squid authentication may also be displayed in this tab.

27.3.17 Wireless Logs

The Wireless logs can be found in the pfSense® webGUI under **Status > System Logs** on the **System/Wireless** tab.

These logs contain entries from the *hostapd* daemon which handles wireless access point connections. This process can be overly verbose when handling client traffic, logging rekeys and other information that can otherwise clutter the main system log.

27.3.18 L2TP Logs

A record of login and logout events is kept on **Status > System Logs**, on the **VPN** tab, under **L2TP Logins**.

Each login and logout is recorded with a timestamp and username, and each login will also show the IP address assigned to the L2TP client. The full log can be found on the **L2TP Raw** tab.

27.3.19 DHCP Logs

The DHCP log view at **Status > System Logs** on the **DHCP** Tab, displays messages and events from the DHCP Daemon and the DHCP client for WANs.

Each DHCP request and reply from DHCP clients is shown here, along with events and errors. IP addresses, MAC addresses, and client-supplied hostnames are all visible in the logs.

See also:

- *Troubleshooting "login on console as root" Log Messages*
- *Troubleshooting "promiscuous mode enabled" Log Messages*
- *Troubleshooting ARP Move Log Messages*

CHAPTER

TWENTYEIGHT

DIAGNOSTICS

These documents cover functions found under the **Diagnostics** menu in pfSense® software.

28.1 DNS Lookup

Diagnostics > DNS Lookup performs simple forward and reverse DNS queries to obtain information about an IP address or hostname, and also to test the DNS servers used by the firewall.

To perform a DNS Lookup:

- Navigate to **Diagnostics > DNS Lookup**
- Enter a **Hostname** or IP address to query
- Click  **Lookup**

The page displays the results of the DNS query along with supporting information and options.

28.1.1 Results

The **Results** panel contains addresses returned by the DNS query along with the record type.

Underneath the results is a table containing the resolution **Timings** per server. This shows how fast each of the configured DNS servers responded to the specified query, or if they never responded.

The **More Information** panel contains links to ping and traceroute functions on the firewall for this host.

28.1.2 Aliases

When performing a DNS lookup, the GUI can also create a *firewall alias* from the results of the query. The name of the alias is the text entered for the DNS query but with `.` characters replaced by _. For example, a DNS lookup for `example.com` results in an alias named `example_com`.

Click  **Add alias** to create an alias containing the results of the query.

If an alias already exists with that name, the button is labeled **Update Alias** instead. That version of the button will replace the contents of the existing alias with the current results of the DNS lookup.

28.2 Editing Files on the Firewall

Diagnostics > Edit File contains a file editor that allows editing and creating files on the pfSense® filesystem.

Enter the filename to edit or create in **Save / Load from path**, or click **Browse** and locate the file. Once the path is filled in, press **Load**. If a new file is being created, type the path and filename to which it will be saved.

Edit or create the text, then press **Save** when finished. For existing files, the contents will be saved. For new files, the file will be created.

> **Warning:** Be careful when choosing a file to edit! It is very easy to edit the wrong file, or break a piece of code, and render the system unusable. Use of this tool is not recommended except under guidance of support or when there is sufficient knowledge to use it without causing unintended side effects.

28.3 Command Prompt

The command prompt, available at **Diagnostics > Command Prompt**, executes shell commands, PHP code, and can download or upload whole files.

> **Warning:** Exercise caution using any of these utilities. Executing commands and PHP code improperly can render the firewall unusable. Use of this tool is not recommended except under the guidance of a support representative or if there is sufficient knowledge on the part of the user.

28.3.1 Execute Shell Commands

To execute a shell command:

- Navigate to **Diagnostics > Command Prompt**
- Enter the command into the **Command** box under **Execute Shell command**
- Click **Execute**

Commands are executed as if they were run from a console command line, and the page prints the results when the command terminates.

> **Warning: Commands must run and then stop or return**.
>
> Commands that run indefinitely, such as `ping` without a count or `tcpdump` without a limit set will never stop or return output, and will be left running indefinitely in the background until they are manually killed.
>
> Interactive commands, such as `vi` will fail similarly, or may exit due to other issues with the terminal being non-interactive.

Previously used commands from this session can be recalled with the « and » buttons. The browser will forget the previous command list once it leaves the page.

28.3.2 Download

To download a file from the firewall filesystem:

- Navigate to **Diagnostics > Command Prompt**
- Enter the full path name in **File to download**
- Click  **Download**

28.3.3 Upload

To upload a file:

- Navigate to **Diagnostics > Command Prompt**
- Click **Browse**
- Locate and select the file on the local client computer
- Click  **Upload**

Note: Uploaded files are placed in `/tmp/` and can then be moved to alternate locations by other functions (such as the **Execute Shell Command** feature).

28.3.4 PHP Execute

This page can also execute PHP code.

- Navigate to **Diagnostics > Command Prompt**
- Type or paste PHP code into the **Execute PHP Commands** text area
- Click **Execute**

The GUI displays the output from the PHP code above the text area, or an error if the it could not run the code.

28.4 Ping Host

The firewall can send ICMP echo reqests, also known as "pings", to hosts over the network to test if they respond, and to measure latency between the firewall and target hosts. A basic ping test can be performed at the console, and a more detailed test is available in the GUI at **Diagnostics > Ping**.

28.4.1 Ping Options

When performing a ping test from the GUI, the following options are available:

Hostname A hostname or IP address to which the firewall will send pings.

IP Protocol The address type to ping when a hostname is entered that has both A (*IPv4*) and AAAA (*IPv6*) records.

Source Address The IP address from which the ping will be sent. This is especially important when testing LAN-to-LAN VPN connectivity. The default choice allows the operating system to automatically select the closest address to the target, based on the routing table.

Maximum Number of Pings The number of ping requests the firewall will send during this test. A higher count will take longer to complete and display results, especially if the target is down. Default is `3`.

Seconds Between Pings The number of seconds to wait between sending ping requests. Default is `1` second.

28.4.2 Ping from the GUI

To perform a ping test from the GUI:

- Navigate to **Diagnostics > Ping**
- Fill in the *Ping Options*

Note: At a minimum the **Hostname** is required.

- Click  **Ping** to start the test
- Wait for the GUI to display the test results

The GUI will display the results of the test automatically once complete. Do not navigate away from the page while the test is running.

28.4.3 Ping from the Console

A simple ping test may also be performed at the console menu, but without the additional options mentioned earlier. See *Ping host* for more information.

- Access the console menu locally or via SSH with an admin-level account (`admin`, `root`, or another privileged account using `sudo`).
- Enter the menu option which corresponds with **Ping Host** (e.g. `7`)
- Press `Enter`
- Enter the IP address or hostname to ping
- Press `Enter` to start the test
- Wait for the test to complete.

  The console outputs the test results in real time, and pauses afterward.
- Press `Enter` to return to the menu

28.5 Halting and Powering Off the Firewall

The firewall may be shut down safely by the **Halt** function available at **Diagnostics > Halt System** or from the console menu.

> **Warning:** The best practice is to **never** cut power from a running system. Halting before removing power is always the safest choice.

Aafter the operating system halts, the device power will also be turned off if that feature is supported by the hardware.

28.5.1 Halt from the GUI

To halt the operating system from the GUI:

- Navigate to **Diagnostics > Halt System**
- Click  **Halt**
- Click **OK** to confirm the action and start the halt process

28.5.2 Halt from the Console

- Access the console menu locally or via SSH with an admin-level account (`admin`, `root`, or another privileged account using `sudo`).
- Enter the menu option which corresponds with **Halt system** (e.g. `6`)
- Press `Enter`
- Enter the `y` to confirm the action
- Press `Enter` to start the halt process

28.6 Rebooting the Firewall

pfSense® software can be rebooted safely and returned to an operational state using the page at **Diagnostics > Reboot System** or the console.

28.6.1 Reboot Methods

The following reboot methods are possible, but available options may be limited depending on the platform and installation options.

Reboot normally Performs a normal reboot in the traditional way. This method is always available.

Reroot Performs a "reroot" style reboot, which is faster than a traditional reboot but does not restart the entire operating system. All running processes are killed, all filesystems are remounted, and then the system startup sequence is run again. This type of restart is much faster as it does not reset the hardware, reload the kernel, or need to go through the hardware detection process.

Reboot into Single User Mode Restarts the firewall into single user mode for diagnostic purposes. The firewall cannot automatically recover from this state, console access is required to use single user mode and reboot the firewall.

This option is not compatible with ARM-based systems.

> **Warning:** Using this option with ZFS will not automatically return to a normal boot and requires manual intervention at the console. See *ZFS Systems*.

> **Warning:** In single user mode the root filesystem defaults to read-only and other filesystems are not mounted. The firewall also does not have an active network connection. This option must only be used under the guidance of a support representative or a FreeBSD user with advanced knowledge.

Reboot and run a filesystem check Reboots the firewall and forces a filesystem check using `fsck`, run five times. This operation can often correct issues with the filesystem on the firewall.

This option is not compatible with ARM-based systems or ZFS.

28.6.2 Reboot from the GUI

To reboot from the GUI:

- Navigate to **Diagnostics > Reboot System**
- Select the *Reboot Method*
- Click  **Submit** to reboot the system immediately

28.6.3 Reboot from the Console

To reboot from the console:

- Access the console menu locally or via SSH with an admin-level account (`admin`, `root`, or another privileged account using `sudo`).
- Enter the menu option which corresponds with **Reboot system** (e.g. `5`)
- Press `Enter`
- Enter the letter which corresponds with the desired *Reboot Method*
- Press `Enter`

Note: The single user mode and filesystem check options require an uppercase letter to be entered to confirm the action. This is necessary to avoid activating the options accidentally. The reboot and reroot options may be entered in upper or lower case.

28.7 Testing a TCP Port

The **Diagnostics > Test Port** page performs a simple TCP port connection test to check if the firewall can communicate with another host. This tests if a host is up and accepting connections on a given port, at least from the perspective of the firewall.

No data is transmitted to the remote host by this test, it only attempts to open a connection and optionally displays the data sent back from the server.

In the default mode the test attempts a simple TCP handshake (SYN, SYN+ACK, ACK), and if the attempt succeeds, it reports the result.

Note: This test does not function for UDP since there is no way to reliably determine if a UDP port accepts connections in this manner.

To perform a test:

- Navigate to **Diagnostics > Test Port**
- Fill in the fields on the page. The **Hostname** and **Port** fields are required, the rest are optional.
- Click  **Test**.

The following options are available on this page:

Hostname This is the IP address or hostname of the target system. This is a required field.

Port This is the TCP port on the target used by the test. This is a required field and must be a valid port number, meaning an integer between `1` and `65535`.

Source Port An optional specific source port for the query. This is unnecessary in most cases.

Remote Text If checked, this option shows the text given by the server when connecting to the port. The server is given 10 seconds to respond, and this page will display all of the text sent back by the server in those 10 seconds. As such, the test will run for a minimum of 10 seconds when performing this check.

Note: Not all daemons will output text to the user on connect, so this may be blank even if the service is working properly. For example, an SMTP server will respond with a welcome message, as will FTP, but an HTTP daemon will not send any text.

Source Address A specific source IP address or IP Alias/CARP Virtual IP from which the query will be sent. The service being tested may require a specific source IP address, network, etc, in order to make a connection.

IP Protocol This option selects either *IPv4* or *IPv6* to control which type of IP address is used when testing a hostname. If the connection is forced to IPv4 or IPv6 and the hostname does not contain a result using that protocol, the test will produce an error. For example if forced to IPv4 and given a hostname that only returns an IPv6 IP address (AAAA record), the test will fail.

28.8 Troubleshooting

```
nc: bind failed: Address already in use
```

The test produces this error if the **Source Port** field is set to a port currently in use by a local daemon on the firewall. Leave **Source Port** blank or pick another unused port.

See also:

To view a list of *ports currently in use*, visit **Diagnostics > Sockets**.

28.9 Traceroute

The traceroute page, located at **Diagnostics > Traceroute**, works like the *traceroute* command found on many platforms. It sends special packets which, as the name implies, trace a route across the network from the pfSense® host to a remote host. A list of hops between hosts will be displayed, along with response times, as long as the intervening hosts support (or don't filter) traffic required for traceroute to work.

- **Host**: A hostname or IP address to which the route will be traced.
- **IP Protocol**: The address type to use when a hostname is entered that has both A (*IPv4*) and AAAA (*IPv6*) records.
- **Source Address**: The IP address from which the trace will be sent. This is especially important when testing LAN-to-LAN VPN connectivity.
- **Maximum number of hops**: The maximum length of the path to trace. The trace will stop if the path cannot be traced completely after this number of hops.
- **Reverse Address Lookup**: When checked, traceroute will attempt to perform a PTR lookup to locate hostnames for hops along the path. Will slow down the process as it has to wait for DNS replies.
- **Use ICMP**: By default, traceroute uses UDP but that may be blocked by some routers. Check this box to use ICMP instead, which may succeed.

The output will be displayed once the trace is complete. The **Stop** button may be pressed at any time to see the current output of the trace if it is still running or stalled.

28.10 Packet Capturing

28.10.1 Selecting the Proper Interface

To perform a packet capture, first determine the location from which the capture must be taken. A packet capture will look different depending upon the chosen interface and in certain scenarios it is better to capture on one specific interface, and in others, running multiple simultaneous captures on different interfaces is preferable.

To use `tcpdump` at the command line, the "real" interface names that go with the friendly names shown in the WebGUI must be known. Visit **Interfaces > Assignments** and make a note of which physical interfaces (e.g. `igb1`), correspond with the friendly interfaces names on the firewall (e.g. WAN). *Real Interfaces vs. Friendly Names* lists common additional unassigned interface names that are present in many firewalls, depending on their configuration.

Table 1: Real Interfaces vs. Friendly Names

Real/Physical Name	Friendly Name
enc0	IPsec, encrypted traffic
ovpnc0 … ovpnc<x>, ovpns0 … ovpns<x>	OpenVPN, encrypted traffic (Clients, Servers)
pppoe0 … pppoe<x>, poes0 … poes<x>	PPPoE WAN, PPPoE Server
l2tp0 … l2tp<x>, l2tps0 … l2tps<x>	L2TP WAN, L2TP Server
lo0	Loopback Interface
pfsync0	pfsync interface – used internally
pflog0	pf logging – used internally

When selecting an interface, start with where the traffic flows into the firewall. For example, if a user is having trouble connecting to a port forward from outside the network, start with the WAN interface since that is where the traffic originates. If a client PC cannot reach the Internet, start with the LAN interface. When in doubt, try multiple interfaces and filter for the IP addresses or ports in question, keeping in mind when NAT will be applied.

28.10.2 Limiting capture volume

When capturing packets, limiting the volume of packets captured is important. However, the limit should not be too low so that all relevant traffic for the problem being troubleshooted is captured. Capture files also consume disk space, which can be a factor on systems with smaller drives. Large captures will also take more time to download, which can be a concern on remote systems with slow WAN upload capacity.

When capturing without filtering on most networks, even for short time frame, huge amounts of data will end up in the capture to dig through when attempting to locate the problem. Display filters in Wireshark can limit which parts of an existing capture file are shown, but filtering appropriately at the time of capture is preferable to keep the capture file size down and to reduce processing time. Filters are discussed later in this chapter.

With an appropriate filter and packet count, capture files can be manageable and contain useful information.

28.10.3 Packet Captures from the WebGUI

The WebGUI offers an easy-to-use front end to `tcpdump` that performs packet captures which can then be viewed or downloaded for deeper analysis in Wireshark. Because of its simplicity, it can only offer a few options for filtering desired traffic. Even with its limitations, it is sufficient for the capturing needs of most users. If the options available in the GUI are too limiting, skip ahead to *Examples of using tcpdump on the command line*.

Getting a Packet Capture

To make a packet capture in the GUI, navigate to **Diagnostics > Packet Capture**.

Configure the options on the page as follows:

Interface The network interface from which packets will be captured. Each assigned interface on the firewall will appear in the list, along with one entry for IPsec, and individual entries for each OpenVPN client and server.

Enable Promiscuous Mode When checked, a capture will include all traffic arriving on the NIC for any destination MAC Address. Without promiscuous mode, only traffic destined for the host or broadcast will be captured. Certain NICs do not handle promiscuous mode well, so this is unchecked by default.

Address Family Limits the capture to only IPv4 or only IPv6 traffic. This is useful when not filtering by IP address.

Protocol Lists common protocols such as TCP, UDP, ICMP, ICMP6, CARP and others, and an exclusion option for each. To limit the capture to one of these protocols, select it from the list. To exclude one of the protocols, choose the option for the protocol prefixed with *Exclude*. The GUI will reject an attempt to submit an invalid combination (e.g.IPv4 only and ICMP6).

Host Address Filters traffic going to or from a specific host or CIDR-masked subnet. Leave the field blank to capture traffic to and from any host. Some logic is possible using this field:

Exclusions Prefix an address with `!` to exclude it from the capture

Boolean AND Join two addresses with a comma (`,`) to capture traffic between only the specified hosts (e.g. `x.x.x.x,y.y.y.y`)

Boolean OR Join two or more hosts with a pipe (`|`) to capture traffic to or from any of the specified hosts. (e.g. `x.x.x.x|y.y.y.y|z.z.z.z`)

Port Fill in a port number to limit the capture to only TCP or UDP matching the specified port as a source or destination port.

Packet Length Sets the size of the packet itself to capture. Usually the full packet is best (`0`), but for captures run over longer periods of time where the headers matter more than the payload of the packets, limiting this to `64` bytes or so will result in a much smaller capture file that may still have adequate data for troubleshooting purposes.

Count Determines how many packets to capture before stopping. If the capture is not limited in any way, bear in mind that this may be "noisy" and this value might need to increase significantly past the default of `100`, such as `1000` or `10000`.

Level of Detail Selects the amount of detail to display in the GUI when viewing a capture. It does **not** change the level of detail in the capture file itself. This value can be changed afterward, click  **View Capture** to display the capture with the new detail level.

Reverse DNS Lookup Causes a reverse DNS lookup to be performed on hosts included in the packet capture. We do not recommend using this option as it will delay the output due to the extra time taken by reverse DNS lookups. Also it is easier to troubleshoot when viewing IP addresses instead of hostnames, and reverse DNS can sometimes be inaccurate.

Click  **Start** to begin capturing packets. The page will display "Packet Capture is running" across the bottom, indicating the capture is in process.

Click  **Stop** to manually end the capture and view the output. If a maximum packet count was specified for the capture, it will stop automatically when that count is reached.

Viewing the Captured Data

The capture output can be viewed in the WebGUI, or downloaded for later viewing in a program such as Wireshark. For more detail on using Wireshark to view a capture file, see *Viewing Packet Capture File* later in this chapter.

When the packet capture page is loaded after a capture has been completed, a  **View Capture** button is presented that will display the packets from the last capture run. Select the **Level of Detail** option before clicking this button to adjust the contents of the display.

Click  **Download Capture** to download this file for later viewing.

The output shown in the **Packets Captured** frame is in standard `tcpdump` style.

28.10.4 Examples of using tcpdump on the command line

The `tcpdump` program is a command line packet capture utility provided with most UNIX and UNIX-like operating system distributions, including FreeBSD. It is also included in pfSense® firewalls, and usable from a shell on the console or over SSH.

It is an exceptionally powerful tool, but that also makes it daunting to the uninitiated user. The tcpdump binary in FreeBSD 10.3 supports 50 different command line flags, limitless possibilities with filter expressions, and its man page, providing only a brief overview of all its options, is nearly 1200 lines long and 67k. After learning to use it, knowledge of how to interpret the data it provides is also necessary, which can require an in-depth understanding of networking protocols.

A comprehensive review of packet capturing and interpretation of the results is outside the scope of this documentation. For those with a thirst for more than basic knowledge in this area, some recommendations are provided in the *Additional References* section. This section is intended to provide an introduction to this topic, and leave the reader with enough knowledge for basic troubleshooting.

tcpdump commonly used flags

The following table shows the most commonly used command line flags with `tcpdump`. Each option will be discussed in further detail in this section.

Table 2: Commonly Used tcpdump Flags

Flag	Description
-i <interface>	Listen on <interface>, .e.g. “-i igb0”
-n	Do not perform reverse DNS resolution on IP addresses
-w <filename>	Save capture in pcap format to <filename>, e.g. “-w /tmp/wan.pcap”
-s	Snap length: Amount of data to be captured from each frame
-c <packets>	Exit after receiving a specific number of packets
-p	Do not put the interface in promiscuous mode
-v	Verbose output
-e	Print link-layer header on each line

-i flag

The `-i` flag specifies the interface on which tcpdump will listen. Use FreeBSD interface names here, such as `igb0`, `em0`, `vmx0`, etc.

-n flag

Do not resolve IP addresses using reverse DNS. When this option is *not* specified, tcpdump will perform a reverse DNS (PTR) lookup for each IP address. This generates a significant amount of DNS traffic in captures displaying large volumes of traffic. Disable this to avoid adding load to DNS servers. We recommend always using `-n` because it eliminates the delay between a packet’s capture and its display that is caused by performing the reverse lookup. Also IP addresses tend to be easier to read and understand than their PTR records. That is a matter of personal preference though, and in familiar environments where the PTR records are known to provide the actual host names of the devices, captures may be run without `-n` to show the hostnames.

Another reason to use `-n`, is to be "sneaky." One means of detecting packet capturing is looking for spikes and patterns in DNS PTR lookups. Skipping the DNS lookup will not cause any extra traffic to be generated in the process.

-w flag

tcpdump allows capture files to be saved in pcap format for later analysis or analysis on another system. This is commonly done from command line only devices like pfSense so the file can be copied to a host running Wireshark or another graphical network protocol analyzer and reviewed there. When saving to a file using `-w`, the frames will not be displayed in the terminal as they otherwise are.

See also:

See *Using Wireshark with pfSense* for more information about using Wireshark with pfSense.

-s flag

By default tcpdump will only save the first 64 bytes of each frame when capturing to a file. This is enough to get the IP and protocol header for most protocols, but limits the usability of capture files. By using the `-s` flag, tcpdump can be told how much of the frame to capture, in bytes. This is called the snap length.

Table 3: Example Uses of tcdump -s

Flag	Description
-s 500	Capture the first 500 bytes of each frame
-s 0	Capture each frame in its entirety

In most cases, using `-s 0` is the best practice when capturing to a file for analysis on another system. The only exception to this is scenarios where a significant amount of traffic must be captured over a longer period of time. If the information being sought is known to be in the header, the default 64 bytes of each frame may be used to get the required information while significantly reducing the size of the resulting capture file.

-c flag

To capture a certain number of frames and then exit, use the `-c` flag. Example usage: `tcpdump` will exit after capturing 100 frames by specifying `-c 100`.

-p flag

Normally when capturing traffic with `tcpdump`, it puts the network interface into promiscuous mode. When not running in promiscuous mode, the NIC only receives frames destined for its own MAC address as well as broadcast and multicast addresses. When switched into promiscuous mode, the interface shows every frame on the wire that arrives at the network interface. In a switched network, this generally has little impact on the capture. In networks where the device is connected to a vswitch also in promiscuous mode, or a hub, using `-p` can significantly limit noise in the capture when the only traffic of interest is to and from the system performing the capture.

-v flag

The `-v` flag controls the detail, or verbosity, of the output. Using more "v" options yields more detail, so use `-v`, `-vv`, or `-vvv` to view even more detail in the output printed to the console. This option does not affect the detail stored in a capture file when using the `-w` switch, but will instead cause the process to report the number of packets captured every 10 seconds.

-e flag

Normally `tcpdump` does not show any link layer information. Specify `-e` to display the source and destination MAC addresses, and VLAN tag information for any traffic tagged with 802.1q VLANs.

Example capture without -e

This capture shows the default output, containing no link layer information:

```
# tcpdump -ni igb1 -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
23:18:15.830706 IP 10.0.64.210.22 > 10.0.64.15.1395: P 2023587125:2023587241(116) ack␣
↪2091089207 win 65535
23:18:15.830851 IP 10.0.64.210.22 > 10.0.64.15.1395: P 116:232(116) ack 1 win 65535
23:18:15.831256 IP 10.0.64.15.1395 > 10.0.64.210.22: . ack 116 win 65299
23:18:15.839834 IP 10.0.64.3 > 224.0.0.18: VRRPv2, Advertisement, vrid 4, prio 0,␣
↪authtype none, intvl 1s, length 36
23:18:16.006407 IP 10.0.64.15.1395 > 10.0.64.210.22: . ack 232 win 65183
5 packets captured
```

Example capture using -e

Here the link layer information is included by using `-e`. Note the source and destination MAC addresses in addition to the source and destination IP addresses:

```
# tcpdump -ni igb1 -e -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
23:30:05.914958 00:0c:29:0b:c3:ed > 00:13:d4:f7:73:d2, ethertype IPv4 (0x0800),␣
↪length 170: 10.0.64.210.22 > 10.0.64.15.1395: P 2023592509:2023592625(116) ack␣
↪2091091355 win 65535
23:30:05.915110 00:0c:29:0b:c3:ed > 00:13:d4:f7:73:d2, ethertype IPv4 (0x0800),␣
↪length 170: 10.0.64.210.22 > 10.0.64.15.1395: P 116:232(116) ack 1 win 65535
23:30:05.915396 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800),␣
↪length 60: 10.0.64.15.1395 > 10.0.64.210.22: . ack 116 win 65299
23:30:05.973359 00:00:5e:00:01:04 > 01:00:5e:00:00:12, ethertype IPv4 (0x0800),␣
↪length 70: 10.0.64.3 > 224.0.0.18: VRRPv2, Advertisement, vrid 4, prio 0, authtype␣
↪none, intvl 1s, length 36
23:30:06.065200 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800),␣
↪length 60: 10.0.64.15.1395 > 10.0.64.210.22: . ack 232 win 65183
5 packets captured
```

tcpdump Filters

On most firewalls running `tcpdump` with no filters will produce so much output that it will prove very difficult to find traffic of interest. There are numerous filtering expressions available that limit the traffic displayed or captured.

Host filters

To filter for a specific host, append `host` and the IP address to the `tcpdump` command. To filter for host `192.168.1.100` use the following command:

```
# tcpdump -ni igb1 host 192.168.1.100
```

That will capture all traffic to and from that host. To only capture traffic being *initiated* by that host, use the `src` directive:

```
# tcpdump -ni igb1 src host 192.168.1.100
```

Similarly, filtering for traffic *destined* to that IP address is possible by specifying `dst`:

```
# tcpdump -ni igb1 dst host 192.168.1.100
```

Network filters

Network filters narrow the capture to a specific subnet using the `net` expression. Following *net*, specify a dotted quad (`192.168.1.1`), dotted triple (`192.168.1`), dotted pair (`192.168`) or simply a number (`192`). A dotted quad is equivalent to specifying `host`, a dotted triple uses a subnet mask of 255.255.255.0, a dotted pair uses 255.255.0.0, and a number alone uses 255.0.0.0.

The following command displays traffic to or from any host with a `192.168.1.x` IP address:

```
# tcpdump -ni igb1 net 192.168.1
```

The next command will capture traffic to or from any host with a `10.x.x.x` IP address:

```
# tcpdump -ni igb1 net 10
```

Those examples will capture all traffic to or from the specified network. The `src` or `dst` keywords may be used the same as with `host` filters to capture only traffic initiated by or destined to the specified network:

```
# tcpdump -ni igb1 src net 10
```

A CIDR mask can also be passed as an argument to `net`:

```
# tcpdump -ni igb1 src net 172.16.0.0/12
```

Protocol and port filters

Narrowing down by host or network frequently isn't adequate to eliminate unnecessary traffic from a capture. Or the source or destination of traffic may not be significant, and all traffic of a certain type should be captured. In other cases, filtering out all traffic of a specific type can reduce noise.

TCP and UDP port filters

To filter on TCP and UDP ports, use the `port` directive. This captures both TCP and UDP traffic using the specified port either as a source or destination port. It can be combined with `tcp` or `udp` to specify the protocol, and `src` or `dst` to specify a source or destination port.

Capture all HTTP traffic

```
# tcpdump -ni igb1 tcp port 80
```

Capture all DNS traffic

Capture all DNS traffic (usually UDP, but some queries use TCP):

```
# tcpdump -ni igb1 port 53
```

Protocol filters

Specific protocols can be filtered using the `proto` directive or by using the protocol name directly. Parameters passed to the `proto` directive can be specified using the IP protocol number or one of the names `icmp`, `igmp`, `igrp`, `pim`, `ah`, `esp`, `carp`, `vrrp`, `udp`, or `tcp`. Because the normal protocol names are reserved words, they must be escaped with one or two backslashes when used with the `proto` directive, depending on the shell. The shell available in pfSense requires two backslashes to escape these protocol names. If a syntax error is returned, check that the protocol name is properly escaped.

The following capture will show all ICMP traffic on the `igb1` interface:

```
# tcpdump -ni igb1 proto \icmp
```

Specifying `carp` for the protocol will capture CARP traffic but it also needs `-T carp` in order to interpret the CARP packets correctly when viewing the output using `tcpdump`. The GUI makes this adjustment automatically when capturing CARP.

The following capture will show all CARP traffic on the `igb1` interface, which can be useful to ensure CARP traffic is being sent and received on the specified interface. It also omits the `proto` keyword, showing that it works on its own:

```
# tcpdump -i igb1 -T carp carp
```

Negating a filter match

In addition to matching specific parameters, a filter match can be negated by specifying `not` in front of the filter expression. When troubleshooting something other than CARP, and its multicast heartbeats are cluttering the capture output, exclude it as follows:

```
# tcpdump -ni igb1 not proto \\carp
```

Combining filters

Any of the aforementioned filters can be combined using `and` or `or`. The following sections provide some examples.

Display all HTTP traffic to and from a host

Display all HTTP traffic to or from `192.168.1.11`:

```
# tcpdump -ni igb1 host 192.168.1.11 and tcp port 80
```

Display all HTTP traffic to and from multiple hosts

Display all HTTP traffic from either `192.168.1.11` or `192.168.1.15`:

```
# tcpdump -ni igb1 host 192.168.1.11 or host 192.168.1.15 and tcp port 80
```

Filter expression usage

Filter expressions must come after every command line flag used. Adding any flags after a filter expression will result in a syntax error.

Incorrect ordering

```
# tcpdump -ni igb1 -T carp carp -c 2
tcpdump: syntax error
```

Correct ordering

```
# tcpdump -ni igb1 -T carp -c 2 carp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 65535 bytes
14:50:07.426993 IP 198.51.100.12 > 224.0.0.18: CARPv2-advertise 36: vhid=11 advbase=1␣
↪advskew=0 authlen=7 counter=5449924379588860810
14:50:08.436849 IP 198.51.100.12 > 224.0.0.18: CARPv2-advertise 36: vhid=11 advbase=1␣
↪advskew=0 authlen=7 counter=5449924379588860810
2 packets captured
78 packets received by filter
0 packets dropped by kernel
```

More on Filters

This section covered the most commonly used `tcpdump` filter expressions, and probably covers all the syntax most users will need. However this barely scratches the surface of the possibilities. There are many documents on the web that cover tcpdump in general and filtering specifically. See *Additional References* at the end of this chapter for links to more resources.

Practical Troubleshooting Examples

This section details an approach we prefer for troubleshooting a few specific problems. There are multiple ways to approach any problem, but packet capturing can rarely be beat for its effectiveness. Examining the traffic on the wire provides a level of visibility into what is actually happening on the network

Port forward not working

In this example, a new port forward is failing to respond to a request from a host on the Internet. The troubleshooting steps outlined in *Port Forward Troubleshooting* offers one way to approach this, but sometimes packet capturing is the only or easiest way to find the source of the problem.

Start from WAN

First, make sure the traffic is getting to the WAN interface. Start a tcpdump session on the WAN interface, and watch for the traffic:

```
# tcpdump -ni igb1 tcp port 5900
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
11:14:02.444006 IP 172.17.11.9.37219 > 10.0.73.5.5900: S 3863112259:3863112259(0) win␣
↪65535 <mss 1260,nop,nop,sackOK>
```

In this case, a packet comes in from the WAN, so it is making it that far. Note that the first part of the TCP handshake, a packet with only SYN set (the `S` shown), is reaching the firewall. If the port forward was working, a SYN ACK (`S.`) packet would be shown in reply to the SYN. With no return traffic visible, it could be a firewall rule or the target system may be unreachable – turned off, not listening on the specified port, host firewall blocking the traffic, etc.

Check Internal Interface

The next step would be to run a tcpdump session on the internal interface associated with the port forward:

```
# tcpdump -ni igb0 tcp port 5900
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb0, link-type EN10MB (Ethernet), capture size 96 bytes
11:14:38.339926 IP 172.17.11.9.2302 > 192.168.30.5.5900: S 1481321921:1481321921(0)␣
↪win 65535 <mss 1260,nop,nop,sackOK>
```

Looking at the internal traffic, the connection left the inside interface and the local IP address was translated correctly. If this local address matches what was expected, then both the port forward and the firewall rule are working properly, and connectivity to the local PC must be confirmed by other means. If no output was displayed, then there is a problem with the firewall rule or the port forward may have been incorrectly defined. For this example, the target system was unplugged.

IPsec tunnel will not connect

`tcpdump` has some awareness of the protocols being used, which can be very helpful in figuring out problems with IPsec tunnels. The next few examples will show how certain error conditions may present themselves when monitoring with `tcpdump`. The IPsec logs are usually more helpful, but this can confirm what is actually being seen by the firewall. For encrypted traffic such as IPsec, packet capturing of the traffic is of less value as the payload of the captured packets cannot be examined without additional parameters, but it is helpful to determine if traffic from the remote end is reaching the firewall and which phases complete.

This first tunnel has an unreachable peer:

```
# tcpdump -ni igb1 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes

19:11:11.542976 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
19:11:21.544644 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
```

This tunnel attempt has a mismatched PSK, notice how it attempts to move to phase 2, but then stops:

```
# tcpdump -ni igb1 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
19:15:05.566352 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
19:15:05.623288 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 1 R agg
19:15:05.653504 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪inf[E]
```

Now Phase 1 is OK but there is a mismatch in the Phase 2 information. It will repeatedly attempt phase 2 traffic but there will not be any traffic in the tunnel:

```
# tcpdump -ni igb1 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
19:17:18.447952 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
19:17:18.490278 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 1 R agg
19:17:18.520149 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
19:17:18.520761 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 2/others R␣
↪inf[E]
19:17:18.525474 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪inf[E]
19:17:19.527962 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪oakley-quick[E]
```

Finally, a fully working tunnel with two-way traffic after Phase 1 and Phase 2 have completed!:

```
#  tcpdump -ni igb1 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on igb1, link-type EN10MB (Ethernet), capture size 96 bytes
21:50:11.238263 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
21:50:11.713364 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 1 R agg
21:50:11.799162 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 1 I agg
21:50:11.801706 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪inf[E]
21:50:11.812809 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 2/others R␣
↪inf[E]
21:50:12.820191 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪oakley-quick[E]
```

(continues on next page)

(continued from previous page)

```
21:50:12.836478 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 2/others R␣
↪oakley-quick[E]
21:50:12.838499 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others I␣
↪oakley-quick[E]
21:50:13.168425 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x1), length␣
↪132
21:50:13.171227 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x1), length␣
↪132
21:50:14.178820 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x2), length␣
↪132
21:50:14.181210 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x2), length␣
↪132
21:50:15.189349 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x3), length␣
↪132
21:50:15.191756 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x3), length␣
↪132
```

Traffic traversing an IPsec tunnel

Traffic can also be observed traversing IPsec tunnels by capturing on the `enc0` interface. This can help determine if traffic is attempting to reach the far end by using the tunnel. All traffic for all IPsec tunnels appears on the `enc0` interface.

In the following example, a host on one side of the tunnel is successfully sending an ICMP echo request (ping) to the far side, and receiving replies:

```
# tcpdump -ni enc0
tcpdump: WARNING: enc0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enc0, link-type ENC (OpenBSD encapsulated IP), capture size 65535 bytes
15:52:46.151098 (authentic,confidential): SPI 0xcd77e085: IP 10.3.0.1 > 10.7.0.1:␣
↪ICMP echo request, id 44640, seq 0, length 64
15:52:46.151814 (authentic,confidential): SPI 0xc0afb14d: IP 10.7.0.1 > 10.3.0.1:␣
↪ICMP echo reply, id 44640, seq 0, length 64
15:52:47.154243 (authentic,confidential): SPI 0xcd77e085: IP 10.3.0.1 > 10.7.0.1:␣
↪ICMP echo request, id 44640, seq 1, length 64
15:52:47.154843 (authentic,confidential): SPI 0xc0afb14d: IP 10.7.0.1 > 10.3.0.1:␣
↪ICMP echo reply, id 44640, seq 1, length 64
```

If traffic was not properly entering the tunnel, no output would be shown. If there is a firewall or internal routing issue on the far side, traffic will appear leaving but nothing will show returning.

Troubleshooting Outbound NAT

For complex environments where Manual Outbound NAT is needed, `tcpdump` can be of great assistance in troubleshooting the Outbound NAT configuration. One good capture to use is to look for traffic with private IP addresses on the WAN interface, as everything on WAN should be have NAT applied and appear to be a public IP address. The following capture will display any traffic with RFC 1918 IP addresses as the source or destination. This will show any traffic that is not matching one of the outbound NAT rules, providing information to help review the Outbound NAT configuration to find the problem:

```
# tcpdump -ni igb1 net 10 or net 192.168 or net 172.16.0.0/12
```

28.10.5 Using Wireshark with pfSense

Wireshark, formerly known as Ethereal, is a GUI protocol analysis and packet capture tool that can be used to view and capture traffic much like `tcpdump`. It is Open Source software, freely available at http://www.wireshark.org/. It can also be used to analyze capture files generated by the pfSense® WebGUI, `tcpdump`, Wireshark, or any other software that writes files in the standard pcap file format.

Viewing Packet Capture File

To view a capture file in Wireshark, start the program and then go to **File > Open**. Locate the capture file, and then click the **Open** button. A file with a `.pcap` extension can also be opened by double clicking on it in Windows, OS X, and many Linux distributions with default settings after the Wireshark installation. A screen similar to Figure *Wireshark Capture View* will be shown in which the data from the capture file is displayed.

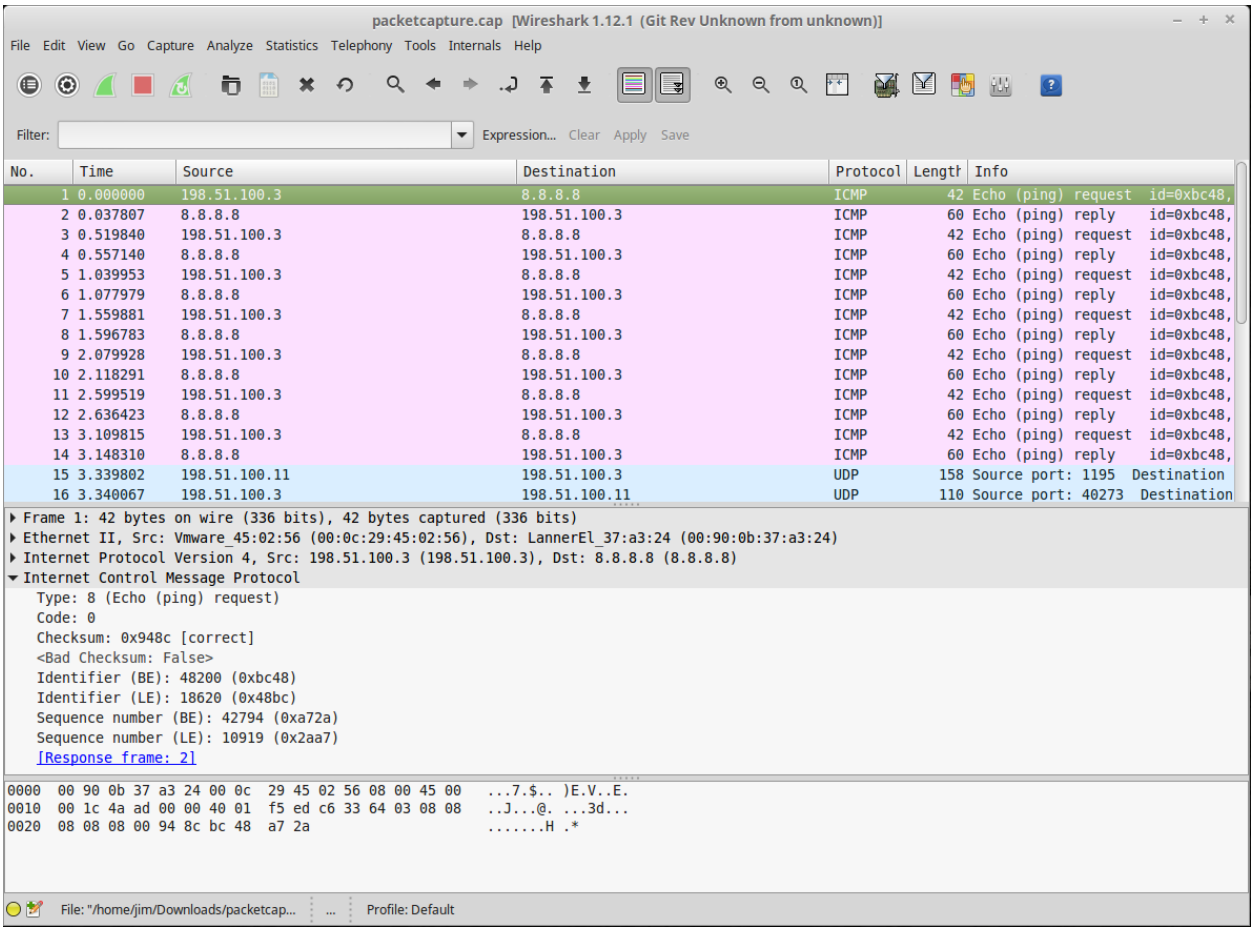


Fig. 1: Wireshark Capture View

As seen in Figure *Wireshark Capture View*, a list summarizing the packets in the capture file will be shown in the top list, with one packet per line. If there are too many, the results can be filtered using the **Filter** box on the toolbar. When a packet is clicked, the lower frames will show the details of what is contained within the packet payload. The first lower pane shows a break-down of the packet's structure, and each of these items can be expanded for more detail. If the packet is of a supported protocol, in some cases it can interpret the data and show even more details. The bottom pane shows a hexadecimal and ASCII representation of the data contained in the packet.

Viewing the capture this way, it is easy to see the flow of traffic with as much or as little detail as needed.

Wireshark Analysis Tools

While some problems will require considerable knowledge of how the underlying protocols function, the analysis tools built into Wireshark helps lessen that need for many protocols. Under the **Analyze** and **Statistics** menus, a few options are present that automate some of the analysis and provide summarized views of what is contained in the capture. The **Expert Info** options under the **Analyze** menu show a list of Errors, Warnings, Notes and network conversations contained in the capture.

Errors may be seen in Wireshark for incorrect checksums. This is because most NICs add the checksum in hardware directly before putting it on the wire. This is the only exception to the earlier note saying what is shown in a packet capture is what is on the wire. Traffic sent out from the system where the capture is taken will have incorrect checksums where they are performed in hardware, though traffic coming in from a remote system should always have correct checksums. Checksum offloading can be turned off to ensure traffic is shown exactly as the host is putting it on the wire, though usually this is something to be ignored. To verify checksums, capture traffic from another system using a network tap or switch span port. Span ports can also be setup on bridges in pfSense, see *Span Port* for more information.

The **Telephony** menu is one example of automated analysis Wireshark can perform to make it easy to see problems with VoIP. In this particular case, VoIP traffic was traversing a MPLS WAN circuit with the provider's routers attached to an OPT interface of pfSense on both sides. A capture from the OPT interface on the initiating end showed no loss, indicating the traffic was being sent to the provider router, but the OPT interface on the opposite end showed considerable packet loss in one direction when multiple simultaneous calls were active. These packet captures helped convince the provider of a problem on their network, and they found and fixed a QoS configuration problem on their side. When viewing a packet capture containing RTP traffic, click **Telephony > RTP > Show all streams** to see this screen.

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
10	13114	192	2244	0x63B69143	ITU-T G.711 PCMU	2103	0 (0.0%)	20.07	0.12	0.01	X
192.1	2244	1	13114	0x6F1D173B	ITU-T G.711 PCMU	1646	477 (22.5%)	179.89	51.24	1.84	X
10	11224	192	2268	0x2247C8D8	ITU-T G.711 PCMU	1321	0 (0.0%)	99.99	5.03	0.07	X
192.1	2268	1	11224	0x6C5B26A1	ITU-T G.711 PCMU	879	460 (34.4%)	340.79	49.96	2.67	X
10	17924	192	2242	0x393CBA89	ITU-T G.711 PCMU	480	0 (0.0%)	20.04	0.15	0.01	X
192.1	2242	1	17924	0x6177246E	ITU-T G.711 PCMU	133	366 (73.3%)	339.79	71.38	9.17	X

Fig. 2: Wireshark RTP Analysis

Remote Realtime Capture

From a UNIX host that has Wireshark available, a realtime remote capture can be run by redirecting the output from an SSH session. This has been tested and known to work on FreeBSD and Ubuntu-based Linux distributions.

In order to use this technique, SSH must be enabled on the pfSense system and an SSH key is required (see *Secure Shell (SSH)*). The key must first be loaded into `ssh-agent` or generated without a passphrase because the redirection will not allow a password or passphrase to be entered. Using `ssh-agent` is the best practice, as any key without a passphrase is very insecure.

Before attempting this technique, check that the user can connect to the pfSense firewall using an SSH key without needing to type the passphrase. The first time the user connects, they are prompted to save the host key, so that must also be done before trying to start wireshark. `ssh-agent` may also be started from a terminal window or shell like so:

```
# eval ssh-agent
Agent pid 29047
# ssh-add
Enter passphrase for /home/jimp/.ssh/id_rsa:
Identity added: /home/jimp/.ssh/id_rsa (/home/jimp/.ssh/id_rsa)
```

Then start an SSH session as usual:

```
# ssh root@192.168.1.1
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.
DSA key fingerprint is 9e:c0:b0:5a:b9:9b:f4:ec:7f:1d:8a:2d:4a:49:01:1b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.1' (DSA) to the list of known hosts.

*** Welcome to pfSense ***
[...]
```

After confirming the SSH connection works, start the remote capture as follows:

```
# wireshark -k -i <(ssh root@192.168.1.1 tcpdump -i igb1 -U -w - not tcp port 22)
```

Replace `192.168.1.1` with the IP address of the pfSense firewall. The `not tcp port 22` filter excludes traffic from the SSH session, which will otherwise clog the capture output. The above is written in BASH style syntax, but may work with other shells. Adjust the `tcpdump` arguments for the interface, and add additional expressions. The `-U` and `-w -` are necessary so that it writes the output to `stdout`, and writes each packet as it arrives.

See also the Capture Setup/Pipes page on the Wireshark wiki for other related techniques.

28.10.6 Additional References

This chapter only scratches the surface of the possibilities with packet captures. Packet capturing is a very powerful means of troubleshooting network connectivity issues, and troubleshooting skills are greatly improved when the possibilities are learned in more depth. The following links are related resources with deeper knowledge beyond the scope of this documentation.

Computer Networking: Internet Protocols in Action by Jeanna Matthews

Tcpdump Filters by Jamie French

Tcpdump Advanced Filters by Sebastien Wains

Tcpdump Filters by Marios Iliofotou

FreeBSD Man Page for tcpdump

Capturing packets is the most effective means of troubleshooting problems with network connectivity. Packet capturing, also known as "sniffing", shows packets "on the wire" coming in and going out of an interface. Observing how traffic is sent and received by the firewall is a great help in narrowing down problems with firewall rules, NAT entries, and other networking issues. This chapter covers obtaining packet captures from the pfSense® WebGUI, with `tcpdump` at the command line in a shell, and using Wireshark.

28.10.7 Capture frame of reference

Keep in mind that packet captures show exactly what is on the wire. A packet capture is the first process to see traffic when receiving packets and last to see traffic when sending packets as they flow through the firewall. It sees traffic before firewall, NAT, and all other processing on the firewall happens for traffic coming into that interface, and after all that processing occurs for traffic leaving that interface. For incoming traffic, captures will show traffic that arrives on an interface on the firewall regardless of whether that traffic will be blocked by the firewall configuration. Figure *Capture Reference* illustrates where tcpdump and also the WebGUI packet capture interface ties into the processing order.

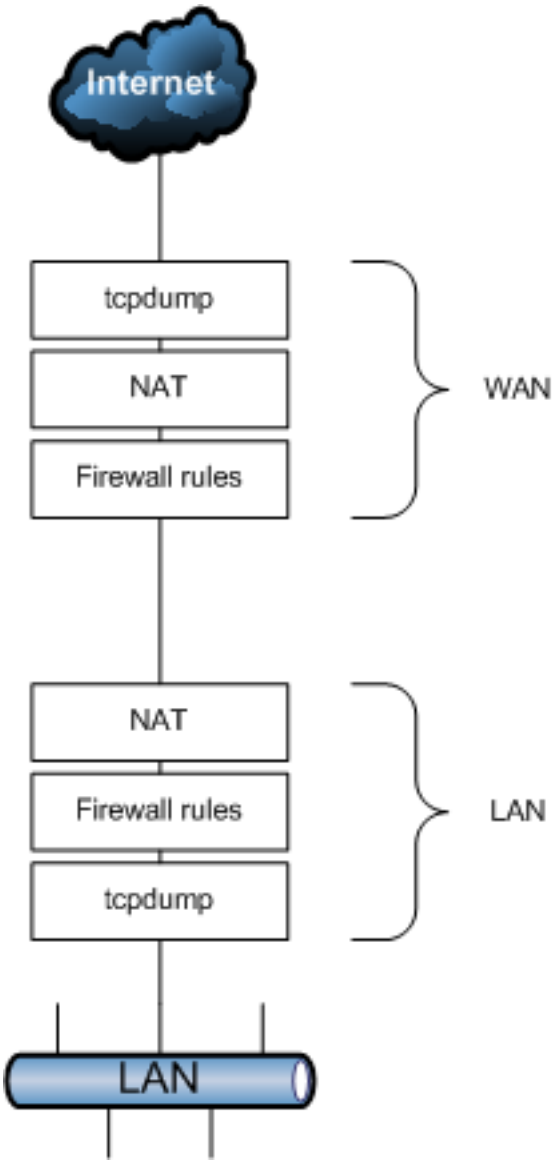


Fig. 3: Capture Reference

CHAPTER

TWENTYNINE

PACKAGES

29.1 Package Manager

Packages are managed at **System > Packages** (Figure *Package Listing*). The listings there, presented in alphabetical order, show all of the information about a package.

Name The name of the package. This is a unique, and typically short, name used to identify the package. On some packages, the name is a link to more information about the package.

Version The version number of the package. This number is specific to the package on pfSense, and is not necessarily related to the version of the underlying software (if there is any). The version number is also a link to recent changes for the package.

Description Longer text describing the package, its purpose, and so on. If the package depends upon other packages, the GUI lists them here denoted by  .

Warning: For security reasons, keep the installed packages to the bare minimum required for a deployment.

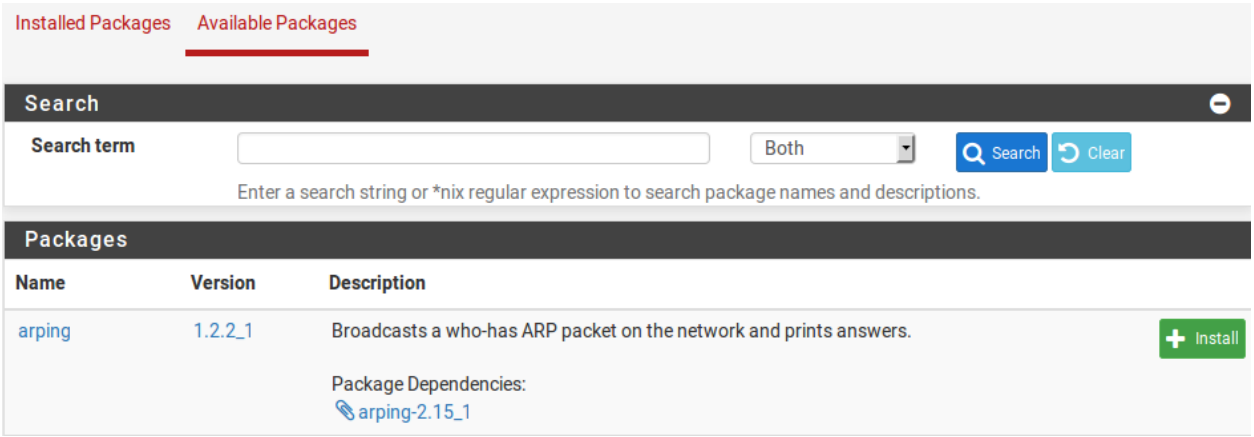


Fig. 1: Package Listing

See also:

- *Troubleshooting Upgrades* (Packages and Updates use the same backend)
- *Troubleshooting DNS Resolution Issues*

- *Troubleshooting Routes*
- *Troubleshooting a Broken pkg Database*

29.1.1 Installing Packages

Packages are installed as follows:

- Navigate to **System > Packages**, **Available Packages** tab
- Locate the package to install in the list

Tip: Search for a package by entering a value in the **Search term** box and clicking **Search**

- Click **Install** to the right of the package entry
- Click **Confirm** to proceed with the package installation

After confirming the installation, the GUI displays the package installation screen containing the install progress (Figure *Post-Install Package Screen*).

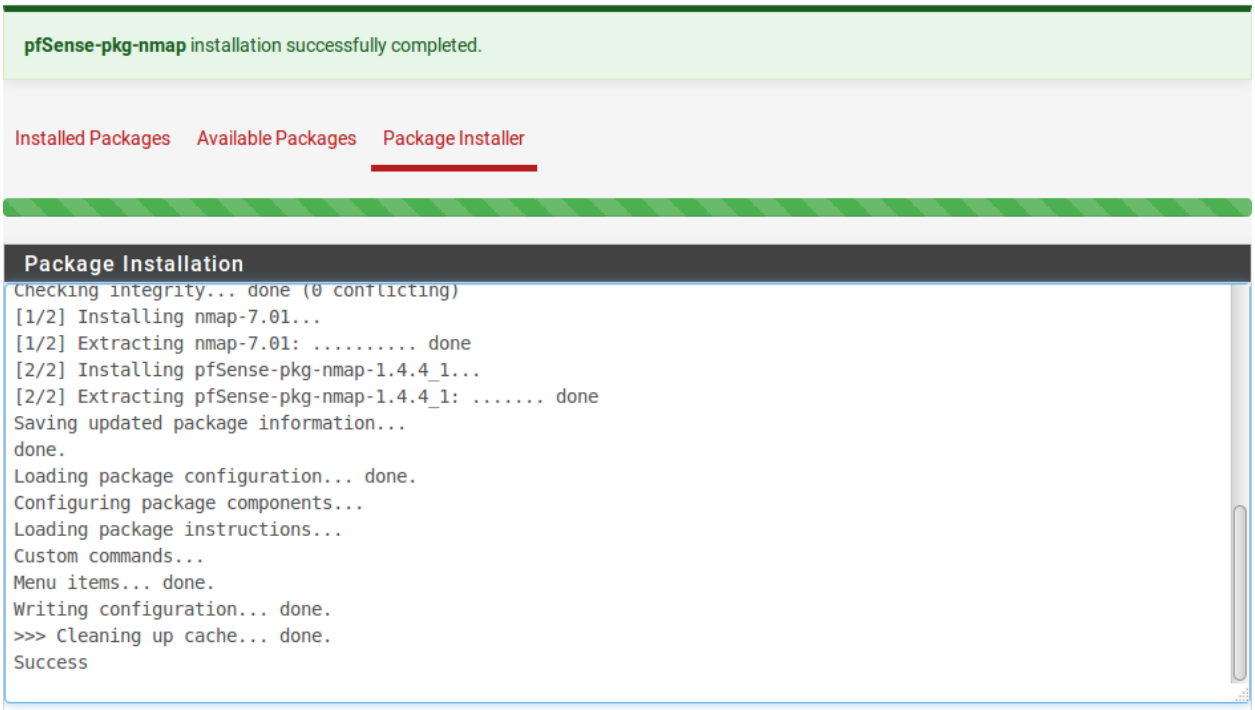


Fig. 2: Post-Install Package Screen

29.1.2 Reinstalling and Updating Packages

Packages are reinstalled and updated the same way they are installed:

- Navigate to **System > Packages**, **Installed Packages** tab

  The list will look like Figure *Installed Package List*

- Locate the package to reinstall or update in the list

  If there is a newer version available than is installed, the **Package Version** column will state the old and new versions with special highlighted text

- Click  to update or  to reinstall the package

- Click  **Confirm** to proceed with the package reinstallation

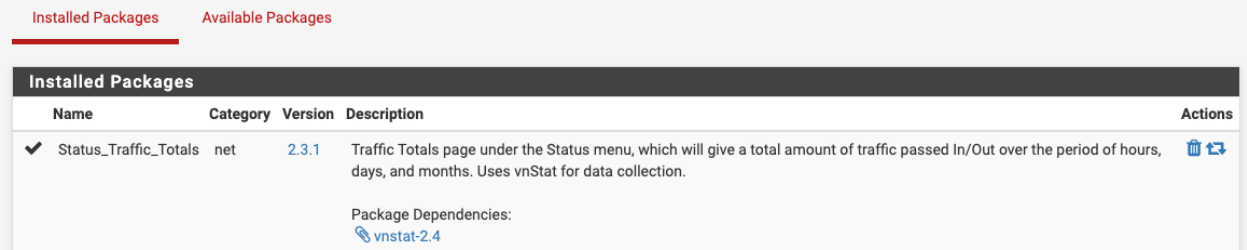


Fig. 3: Installed Package List

29.1.3 Uninstalling Packages

To uninstall a package:

- Navigate to **System > Packages**, **Installed Packages** tab
- Locate the package to uninstall in the list
- Click  to remove the package
- Click  **Confirm** to proceed with the package removal

29.2 Package List

The following packages are available from the pfSense® software package repository.

Warning: Packages availability can change over time. Check **System > Package Manager > Available Packages** for an always up-to-date list of packages.

Tip: The package name in the list below links to documentation for the package, if it exists.

ACME The Automated Certificate Management Environment (ACME) package manages certificates from ACME providers such as Let's Encrypt.

See also:

ACME package

arping Broadcasts a `who-has` ARP packet on the network and prints answers.

See also:

Arping Package

arpwatch Monitors devices on directly attached networks and notifies when it detects new MAC addresses.

apcupsd Controls all APC UPS models. It can monitor and log the current power and battery status, perform automatic shutdown, and can run in network mode to power down other hosts over the network.

aws-wizard **(pfSense Plus Only)** AWS VPC VPN Connection Wizard. Automatically creates a VPN tunnel and BGP configuration to communicate with an Amazon AWS VPC.

Avahi Facilitates service discovery on a local network via the mDNS/DNS-SD protocol suite. This enables clients to plug a laptop or computer into a network and instantly be able to view other people who they can chat with, find printers to print to or find files being shared. In addition it supports mDNS reflection across LAN segments. Compatible technology is found in Apple MacOS X (branded Bonjour and sometimes Zeroconf).

See also:

Avahi package

Backup Backs up and restores arbitrary files and directories.

See also:

Backup Files and Directories with the Backup Package

bandwidthd Tracks TCP/IP network usage and creates graphs of data consumption for individual IP addresses.

BIND Provides a GUI for BIND DNS server.

cellular Provides a GUI for cellular cards (e.g. 3G/4G/LTE), it currently supports certain Huawei models.

Cron Manages scheduled commands run periodically by the firewall.

Darkstat A network statistics gatherer that offers bandwidth graphs for an interface, as well as traffic to/from specific IP addresses. Once installed, it appears under **Diagnostics > darkstat**.

filer Stores custom files persistently in the configuration.

FreeRADIUS A free implementation of the RADIUS protocol, used for Authentication, Authorization, and Accounting (AAA).

See also:

FreeRADIUS package

FRR A GUI for the FRR routing daemon which supports BGP, OSPF, and OSPF6.

FTP Client Proxy A basic FTP client proxy using `ftp-proxy` from FreeBSD.

HAproxy A reliable, high performance TCP/HTTP(S) load balancer. This package implements the TCP, HTTP and HTTPS balancing features from haproxy and supports ACLs for smart backend switching. A good replacement when relayd is incapable of handling load balancing needs. Requires SSD/HDD.

See also:

HAProxy

HAproxy-devel The development package for *HAproxy*.

iperf A tool for testing network throughput, loss, and jitter. Can act as a client or a server.

See also:

iperf package

ipsec-profile-wizard **(pfSense Plus Only)** Creates IPsec configuration profiles for Apple devices (iOS and OS X) and IPsec import script bundles for Windows devices.

LADVD Sends and decodes link layer advertisements.

Supports LLDP (Link Layer Discovery Protocol), CDP (Cisco Discovery Protocol), EDP (Extreme Discovery Protocol) and NDP (Nortel Discovery Protocol).

See also:

Using LLDP on pfSense software

LCDproc LCD display drivers and service.

Lightsquid A high performance web proxy reporting tool. Includes realtime proxy statistics (SQStat). Requires the **Squid** package. Requires SSD/HDD.

lldpd Provides support for the 802.1ab Link Layer Discovery Protocol (LLDP), as well as support for several proprietary discovery protocols including Cisco Discovery Protocol (CDP), Extreme Discovery Protocol (EDP), Foundry Discovery Protocol (FDP), and Nortel Discovery Protocol (NDP / SONMP).

Similar to *LADVD* but a more modern implementation.

Mailreport Manages periodic e-mail reports containing command output and log file contents.

MTR An enhanced traceroute replacement. `mtr` combines the functionality of the traceroute and ping programs in a single network diagnostic tool.

Netgate Firmware Upgrade (pfSense Plus Only) Provides a mechanism to update firmware on certain Netgate hardware models. Varies by hardware and may be Coreboot, Blinkboot, or other types of firmware.

net-snmp The NET-SNMP implementation of SNMP. More extensible than the built-in SNMP daemon (bsnmpd), and supports SNMPv3 authentication and TLS encryption.

nmap A utility for network exploration and security auditing. It supports scanning to determine active hosts, many port scanning techniques to determine services offered by hosts, version detection to determine what application/service is running on a port, and TCP/IP fingerprinting to identify the OS on remote hosts. It also offers flexible target and port specification, decoy/stealth scanning, SunRPC scanning, and more.

See also:

Nmap package

node_exporter Prometheus exporter for machine metrics.

Notes Maintains a list of noteworthy items for the system.

NRPE Provides a GUI for Nagios NRPE. It execute Nagios plugins on remote hosts and report the results to the main Nagios server.

It also allows Nagios to execute plugins like `check_disk`, `check_procs`, etc. on remote hosts.

ntopng A network probe that shows network usage in a way similar to what top does for processes. In interactive mode, it displays the network status on the user's terminal. In Web mode it acts as a Web server, creating an HTML dump of the network status. It sports a NetFlow/sFlow emitter/collector, an HTTP-based client interface for creating ntop-centric monitoring applications, and RRD for persistently storing traffic statistics. Requires SSD/HDD.

Network UPS Tools (NUT) Provides support for monitoring of Uninterruptible Power Supplies. It supports UPS units attached locally via USB or serial, and remote units via the SNMP protocol, the APCUPSD protocol or the NUT protocol.

See also:

Nut package

Open-VM-Tools A suite of open source utilities which enhance the performance of VMware virtual machine guest operating systems and improve management of virtual machines.

See also:

Open VM Tools package

OpenVPN Client Export Generates pre-configured OpenVPN configuration files for clients, Windows Client installers with configurations bundled, and Mac OS X Viscosity configuration bundles, among others.

See also:

OpenVPN Client Export Package

OpenVPN Client Import **(pfSense Plus Only)** Imports a unified OpenVPN client configuration file as exported by an OpenVPN server, allowing clients to be easily configured without creating a client instance and adding settings manually.

pfBlockerNG Utility for controlling connections through the firewall based on more general criteria than firewall rules (e.g. by country, by domain name, etc). Manages IPv4/v6 List Sources into 'Deny, Permit or Match' formats. GeoIP database by MaxMind Inc. (GeoLite2 Free version). De-Duplication, Suppression, and Reputation enhancements. Provision to download from diverse List formats. Advanced Integration for Proofpoint ET IQRisk IP Reputation Threat Sources. Domain Name (DNSBL) blocking via Unbound DNS Resolver.

See also:

pfBlocker-NG Package

pfBlockerNG-devel The development version of pfBlockerNG

See also:

pfBlocker-NG Package

PIMD A GUI for `pimd`, a multicast routing daemon. Primarily replaces the role of the built-in IGMP Proxy function to allow routing multicast traffic across multiple interfaces. Not a replacement for Avahi.

Routed A RIP v1 and v2 daemon.

> **Warning:** This package has been deprecated.

RRD Summary Gives a total amount of traffic passed In/Out during this and the previous month. Set to be replaced by the **Traffic totals** package.

Service Watchdog Monitors for stopped services and restarts them.

Shellcmd Manages boot-time commands.

See also:

Executing Commands at Boot

Siproxd A proxy for handling multiple SIP devices using a single public IP address.

See also:

Siproxd package

snmptt SNMP Trap Translator for use with the Net-SNMP. Easy to setup and use.

Snort An open source network intrusion detection and prevention system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. SSD/HDD is strongly recommended.

See also:

IDS / IPS

Softflowd A flow-based network traffic analyzer capable of Cisco NetFlow data export. Tracks traffic flows and reports via NetFlow to a collecting host.

See also:

Exporting NetFlow with softflowd

Squid A high performance web proxy cache. It combines Squid as a proxy server with its capabilities of acting as a HTTP/HTTPS reverse proxy. It includes an Exchange-Web-Access (OWA) Assistant, SSL filtering and antivirus integration via C-ICAP. SSD/HDD recommended.

See also:

Squid

SquidGuard A high performance web proxy URL filter. SSD/HDD recommended.

See also:

Configuring the SquidGuard Package

Status Traffic Totals Calculates a total amount of traffic passed In/Out over the period of hours, days, and months. Uses vnStat for data collection. It shows up in the menu under **Status > Traffic Totals**.

See also:

Status Traffic Totals

Stunnel A TLS encryption wrapper between a remote client and local or remote servers.

See also:

Stunnel package

Sudo Delegates privileges to users in the shell so commands can be run as other users, such as `root`.

See also:

Sudo Package

Suricata A high performance network IDS/IPS and security monitoring engine by OISF. SSD/HDD strongly recommended.

Syslog-ng A modern syslog server which supports TCP and TLS encryption, among other features.

Note: This service is not intended to replace the default syslog server on the firewall but rather acts as an independent syslog server.

System Patches Manages custom code patches to be applied and maintained to the system. These can be commits from Github, manual diffs, or loaded from URLs.

See also:

System Patches Package

Telegraf An agent written in Go for collecting, processing, aggregating, and writing metrics.

TFTPD GUI for a TFTP server, using the versatile **tftp-hpa** daemon.

Tinc A Virtual Private Network (VPN) daemon that uses tunneling and encryption to create a secure private network between hosts on the Internet. A single tinc daemon can accept more than one connection at a time, thus making it possible to create larger virtual networks, because some limitations are circumvented. Instead of most other VPN implementations, tinc encapsulates each network packet in its own UDP packet, instead of encapsulating all into one TCP or even PPP over TCP stream. This results in lower latency, less overhead, and in general better responsiveness and throughput.

***WireGuard*®** WireGuard is a new VPN Layer 3 protocol designed for speed and simplicity. It performs nearly as fast as hardware-accelerated IPsec and has only a small number of options in its configuration.

> **Warning:** This package is **experimental**.

Zabbix-agent Zabbix Monitoring agent. The agent gathers operational information locally and reports data to Zabbix server for further processing. The agent can also generate alerts in case of failures. Available in multiple versions.

Zabbix-proxy Zabbix Agent proxy. Collects performance and availability data on behalf of the Zabbix server, lowering the burden on the server. Available in multiple versions.

The package system in pfSense® software provides the ability to extend the functionality of the software without adding bloat and potential security vulnerabilities to the base distribution. To see the packages available for the current firewall platform being utilized, browse to **System > Packages**, on the **Available Packages** tab.

Some packages have been written by the pfSense community, and others directly developed by Netgate. The available packages vary quite widely, and some are more mature and well-maintained than others. There are packages which install and provide a GUI interface for third-party software, such as Squid, and others which extend the functionality of pfSense software, like the OpenVPN Client Export Utility package which automatically creates OpenVPN configuration files.

Some other examples of available packages (which are not Squid related) are:

- Additional filtering functionality (pfBlockerNG)
- IDS/IPS software (Snort and Suricata)
- Additional VPN technologies (WireGuard, Tinc)
- Bandwidth monitors that show traffic by IP address such as ntopng, and Darkstat.
- Extra services such as FreeRADIUS and BIND.
- Web client proxy (Squid) and proxy filtering (SquidGuard)
- Reverse proxies for HTTP and HTTPS such as HAProxy
- Proxies for other services such as SIP and FTP
- System utilities such as NUT for monitoring a UPS.
- Popular third-party utilities such as nmap, iperf, and arping.
- BGP Routing, OSPF routing, Cron editing, Zabbix agent, and many others.
- Features that were formerly in the base system but were moved to packages, such as RIP (routed)

As of this writing there are more than 60 different packages available; too many to cover them all in this documentation! The full list of packages that can be installed on a particular system is available from within the GUI at **System > Packages** on the **Available Packages** tab.

The packages screen takes longer to load than other pages in the web interface because the firewall fetches package information from the Netgate package repository servers before the page is rendered to provide the most up-to-date

information. If the firewall does not have a functional Internet connection including DNS resolution, this will fail. This is usually caused by a missing or incorrect DNS server configuration, or a missing default gateway.

See also:

- *Troubleshooting Upgrades* (Packages and upgrades both use `pkg`)
- *Troubleshooting DNS Resolution Issues*
- *Troubleshooting Routes*
- *Troubleshooting a Broken pkg Database*
- *Developing Packages*

29.3 ACME package

Let's Encrypt is an open, free, and completely automated Certificate Authority from the non-profit Internet Security Research Group (ISRG). The goal of Let's Encrypt is to encrypt the web by removing the cost barrier and some of the technical barriers that discourage server administrators and organizations from obtaining certificates for use on Internet servers, primarily web servers. Most browsers trust certificates from Let's Encrypt. These certificates can be used for web servers (HTTPS), SMTP servers, IMAP/POP3 servers, and other similar roles which utilize the same type of certificates.

The ACME Package for pfSense® software interfaces with Let's Encrypt to handle the certificate generation, validation, and renewal processes.

Certificates from Let's Encrypt are domain validated, and this validation ensures that the system requesting the certificate has authority over the domain in question. This validation can be performed in a number of ways, such as by proving ownership of the domain's DNS records or hosting a file on a web server for the domain.

By using a certificate from Let's Encrypt for a web server, including a firewall running pfSense software, the browser will trust the certificate and show a green check mark, padlock, or similar indication. The connection will be encrypted without the need for a client to manually trust an invalid or self-signed certificate.

Let's Encrypt certificates are valid for a period of 90 days, so they must be renewed periodically. The ACME package automates this renewal by using a cron job to check once per day to see if a certificate needs to be renewed.

29.3.1 ACME Overview

Rate Limits

Let's Encrypt enforces rate limitations when using the production validation system, such as:

- Five validation failures per account, per hostname, per hour
- Each certificate may have at most 100 SAN entries
- Only 50 certificates may be created per domain per week

A testing validation system exists for developers who are programming clients or administrators testing their settings. The test system has higher limits, which aid testing and development, but the test system does not produce certificates which are trusted publicly.

Security Limitations

When validating using a method such as webroot or standalone the service must be available to the Internet on its standard port: 80 for HTTP or 443 for TLS-ALPN. This is a security limitation to prevent a user from running an alternate web server on a high- numbered port and obtaining a certificate for a server they do not normally control.

Validation Process

When creating a certificate, one or more fully qualified domain names (FQDNs) are listed on the certificate in the SAN list. Let's Encrypt will query each of these domain names in DNS in different ways depending on the validation method.

When a validation method starts, the client obtains an authorization value from the server (authz).

For DNS-based methods, Let's Encrypt checks for a TXT record in the form of `_acme-challenge.<domain name>` which must contain the authorization value. This proves that the person or system requesting the certificate controls DNS records for the domain.

For File-based methods such as webroot or standalone, Let's Encrypt connects to an IP address obtained by resolving the A record for the FQDN and requests a file from the web server at `.well-known/acme-challenge/` underneath the webroot directory. This file contains the authorization value. This proves that the person or system requesting the certificate controls web server for the domain name.

29.3.2 Obtaining a Certificate

These instructions cover the general process of obtaining a certificate. Specific settings will vary by deployment, and each section below links to the settings for each area.

Generate an Account Key

Before a certificate can be created by the firewall, the firewall must first obtain an account key. This key is typically unique for each server, but can be shared.

For users unfamiliar with Let's Encrypt, the first key should be for the staging system which has no rate limits but is not valid for public use. Once a certificate is successfully issued by the staging system, create an account key for the production system and then issue the certificate again using that key.

To create and register an account key:

- Navigate to **Services > ACME Certificates**, **Account Keys** tab
- Click  **Add**
- Fill in the info as described in *Account Key Settings*
- Click  **Create new account key**
- Click  **Register ACME account key**
- Click **Save**

Create a certificate

The next step is to create a certificate entry.

- Navigate to **Services > ACME Certificates**, **Certificates** tab
- Click  **Add**
- Fill in the info as described in *Certificate Settings*
- Add one or more **Domain SAN List entries** (*Certificate Settings*) with appropriate validation settings (*Validation Methods*)
- Add one or more **Actions list** entries (*Certificate Settings*)
- Click **Save**

Configure General Settings

The last step is to enable at least the **Cron Entry** to ensure that the ACME package will automatically renew certificates before they expire. See *General Settings* for detailed descriptions of the options.

- Navigate to **Services > ACME Certificates**, **General Settings** tab
- Check **Cron Entry**
- Check **Write Certificates** (optional)
- Click **Save**

29.3.3 ACME Package Settings

These sections describe the settings for each tab in the ACME package.

Account Key Settings

An ACME account key has the following settings:

Name A short name for the key

Description A longer string describing the key

ACME Server The ACME server to which this key will be registered by the package.

Currently supported options are:

Let's Encrypt Staging ACMEv2 Use this server when testing the certificate validation process. Does not produce publicly trusted certificates.

Let's Encrypt Production ACMEv2 Use this server for trusted production certificates.

BuyPass Production ACMEv2 An alternative service for ACME certificates

E-Mail Address An e-mail address which Let's Encrypt will use to send certificate expiration notices if certificates are not renewed in a timely manner.

Account Key The RSA private key for this entry. To create a new key, click  Create new account key.

Certificate Settings

Certificate entries have the following settings:

Name A short name for the certificate

Description A longer string describing the certificate

Status Whether or not this entry is active

Active This entry will be processed manually and by the Cron job (*General Settings*)

Disabled This entry will be ignored

Acme Account The account key ACME will use when requesting the certificate (see *Generate an Account Key*)

Private Key The key length of the private key for this certificate. May be either RSA or ECDSA in several pre-defined sizes. Select *Custom* to manually enter a private key generated elsewhere

2048-bit RSA is an acceptable default choice, but larger keys are more secure

OCSP Must Staple When set, ACME will configure the certificate request for OCSP Stapling

> **Warning:** Do not enable this option unless all consumers of the certificate support OCSP Stapling.

Domain SAN List A list of all domain names which will be included in this certificate as Subject Alternative Name (SAN) entries.

Note: A certificate can contain up to 100 SAN entries, and they can use the same or different update methods. Each SAN must be individually validated by Let's Encrypt before a certificate will be issued.

Mode Whether or not this SAN is active in the certificate

Domain Name The domain name for a SAN entry in this certificate (e.g. `www.example.com`)

Method The method used by ACME to validate ownership of this domain. Method settings are described in (*Validation Methods*)

Click  **Add** for additional SAN entries

DNS Providers also have some common settings which appear for all types:

DNS Alias An alternative domain name used by the validation process. Instead of updating the DNS record for **Domain Name** directly, the package uses this domain name is used instead. See *DNS Alias Mode* for details.

DNS Alias Mode When set, controls whether or not the DNS alias mode used is Challenge Alias (Unchecked, Default) or Domain Alias (Checked). See *DNS Alias Mode* for details.

DNS-Sleep The amount of time the ACME validation process will wait after making DNS changes before attempting to validate. Some DNS services take a few minutes to propagate entries after making backend changes.

The default settings are typically sufficient, but slower providers may require a longer sleep time.

Actions List Commands to run after the package renews a certificate.

Mode Whether or not this action is active

Command Full path to command and arguments, service name, or name of script

Method Defines how the **Command** is executed by the package

Shell Command The **Command** is a full path to a shell command and its arguments

PHP Command Script The **Command** value is run as PHP code

Restart Local Service The name of a local service to restart

Restart Remote Service The name of a remote service to restart via XMLRPC. This utilizes the system *XMLRPC sync configuration*

The GUI includes several examples of common actions

Certificate Renewal After When the package will attempt a renewal for the certificate. Default is `60` days (2 months). Certificates are valid for a maximum of `90` days.

Validation Methods

ACME providers can validate by checking the contents of a TXT record in DNS, or by fetching a file in a known location from a web server.

The ACME package support validating directly with standalone methods or webroot, but those options are less secure than DNS-based options. The ACME package also supports numerous methods to update various DNS providers. Wildcard certificates can only be obtained through DNS-based methods (*Wildcard Certificates*)

Tip: DNS-based update methods are the best practice as they does not require external inbound access. They can be used for internal systems that do not allow or cannot receive Internet traffic.

The following list is only a portion of the validation methods supported by the package.

See also:

DNS methods also have a common option for DNS Alias mode. See *Certificate Settings* and *DNS Alias Mode* for details.

nsupdate

The `nsupdate` method uses RFC 2136 style DNS updates to populate a TXT record in DNS.

Before starting, an appropriate DNS key and settings must be in place in the DNS infrastructure for the domain to allow the host to update a TXT DNS record for `_acme-challenge.<domain name>`.

This method has the following options:

Server The IP address or hostname of the DNS server to which the client sends updates

Key Name The name of the update key

Leave this blank unless it is different than `_acme-challenge.<domain name>`

Key Algorithm The algorithm used for the key, which must match the key and the server

Key The update key for this record

Zone Sets the zone name the package sends to the DNS server in the update request

DNS-Manual

The manual DNS method can be utilized when a firewall cannot receive inbound traffic and it does not have access to any automatic DNS-based method.

The **manual** in the name indicates that the process **must be performed by hand** both initially and when it is time to renew the certificate. The firewall obtains the authorization value and then the TXT record must be manually created or updated with this value.

Warning: Avoid using this method unless no other method is available.

To use this method:

- Add an entry to the **Domain SAN list**
- **Mode**: Enabled
- Enter domain name (e.g. `myhost.example.com`)
- Set **Method** to *DNS-Manual*
- Click **Save**
- Click **Issue**
- Locate the record info in the output:

```
[Mon Feb 6 14:49:23 EST 2017] Add the following TXT record:
[Mon Feb 6 14:49:23 EST 2017] Domain: '_acme-challenge.www.example.com'
[Mon Feb 6 14:49:23 EST 2017] TXT value: 'xPrykHSri5epT5yrJJWyY536Z1T51r_
↪Ef4LkWJry-iw'
[Mon Feb 6 14:49:23 EST 2017] Please be aware that you prepend _acme-challenge.␣
↪before your domain
[Mon Feb 6 14:49:23 EST 2017] so the resulting subdomain will be: _acme-challenge.
↪www.example.com
[Mon Feb 6 14:49:23 EST 2017] Please add the TXT records to the domains, and␣
↪retry again.
```

- Add or update the TXT record in the domain's DNS server for `_acme-challenge.<domain name>` with the TXT value from the output
- Wait approximately 2 minutes, or longer, for DNS to propagate
- Click **Renew**

Namecheap API

For certain accounts with Namecheap, API access may be obtained that allows remote manipulation of DNS records. This can be used with the ACME package to validate certificates for domains with DNS hosted at Namecheap using their BasicDNS servers.

Warning: The Namecheap DNS API requires that the client read all records and then write them all back when making any change. This is potentially dangerous. Take a backup of all DNS records on the domain before attempting to use the API.

The first step is to request API access:

- Login to a Namecheap account
- Navigate to **Profile > Tools** under the account
- Look for **Namecheap API Access** under **Business & Dev Tools**
- If the status does not say **On**, then click **Manage** and change the slider to **On**.

Note: API access must be approved by Namecheap. There are qualifications to meet, such as a specific number of domains or a balance on the account. Check the Namecheap API documentation for more information. The process is documented as taking 2 days, but may take longer. If API access is not enabled after several days, contact Namecheap support.

Once the API is enabled, then perform the following steps:

- Login to a Namecheap account
- Navigate to **Profile > Tools** under the account
- Look for **Namecheap API Access** under **Business & Dev Tools**
- Click **Manage**
- Note the API key for use in the ACME package
- Click **Edit** and add whitelisted IP addresses that can contact the API using this API key.

Now setup the account in the ACME package:

- Add an entry to the **Domain SAN list**
- **Mode**: Enabled
- Enter domain name (e.g. `myhost.example.com`)
- Set **Method** to *DNS-Namecheap*
- Click + to expand the method-specific settings
- Fill in the info

  API Key The API Key displayed in the Namecheap API Access manager, as described previously.

  Username The Namecheap account username associated with the API Key.
- Ensure the other options are set properly, per *Create a certificate*.
- Click **Save**
- Click **Issue/Renew**

Other DNS Methods

The package contains several additional DNS-based methods for other providers. These work similar to the nsupdate method above, but have configuration values specific to each provider. Contact the DNS provider or server administrator to obtain the necessary settings or credentials.

FTP Webroot

The **FTP webroot** method is useful when the firewall is performing NAT (port forward or 1:1) or reverse proxy duty for handling traffic for the domain. The firewall can use SFTP or FTPS to store the domain validation files on a web server behind the firewall so it does not have to host the files itself.

We recommend using this method when no DNS update method is available for use by the firewall.

This method has the following options:

Server The server where the package will send the challenge response files, e.g. `sftp://x.x.x.x`

> **Note:** This method supports supports `sftp://` and `ftps://` servers.

Username/password Credentials for the SFTP/FTPS account

Folder **Full path** to the target directory including `/.well-known/acme-challenge` at the end

> **Warning:** Make sure the specified user has write permissions to the directory!

Webroot Local Folder

This method works similar to FTP Webroot but with the files hosted on the firewall itself. This method cannot be utilized by the WebGUI web server as that would mean exposing the GUI to the Internet, which is a major security issue.

This method can, however, be used in conjunction with the HAProxy package to host the files on the firewall itself in some circumstances. See https://forum.netgate.com/post/677786 for details.

Standalone (HTTP/TLS-ALPN)

The **Standalone** methods for HTTP and TLS-ALPN run a small web server natively that is active only while the validation process is running. The TLS-ALPN method is more secure as it encrypts communication with the ACME provider.

> **Warning:** We do not recommend using these methods as they expose a service on the firewall to the Internet. Only use these methods if no other method is available.

> **Warning:** The service **must** be accessible using port `80` (HTTP) or `443` (TLS-ALPN)!

If the firewall is using port 80 (HTTP) or 443 (TLS-ALPN) for another service, such as the firewall GUI or its redirect, then this method may not be viable. If the service on the port is public, then it cannot be used. If the service is private, then it may be possible to relocate the existing service or bind the update method to an alternate port, then port forward on the WAN interface.

A firewall rule must allow traffic to the target port at all times, it cannot be automatically enabled and disabled in the current package.

Note: The standalone binding should only be changed if the port is forwarded via NAT to a different port (e.g. 80 forwarded to 8080)

Standalone HTTP

Standalone HTTP Server has the following options:

Port Port to which the package will bind listening for HTTP requests with a stand-alone server. Must be 80 or port 80 must be forwarded to this port on the default gateway WAN.

> **Warning:** If port 80 is used by the standalone service, the GUI redirect must be disabled on **System > Advanced** using the **Disable webConfigurator redirect rule** option. If the redirect is active when standalone mode attempts to use the port, it will print an error message stating that socat is unable to bind to the port.

Bind to IPv6 instead of IPv4 If the domain name for the firewall has both an A and AAAA DNS record, check this option so that validation can occur over IPv6.

Standalone TLS-ALPN

Standalone TLS-ALPN Server has the following options:

Port Port to which the package will bind listening for TLS-ALPN requests with a stand-alone server. Must be 443 or port 443 must be forwarded to this port on the default gateway WAN.

> **Warning:** If port 443 is used by the standalone service, the GUI must be moved to an alternate port on **System > Advanced** using the **TCP Port** option for the GUI. If the redirect is active when standalone mode attempts to use the port, the standalone service will fail.

DNS Alias Mode

DNS Alias mode allows a DNS update method to update an alternate domain name instead of updating a record for the domain name directly.

If the main DNS provider does not support updating TXT records, a CNAME record can point to an alternative domain which does.

Challenge Alias

In Challenge Alias mode (default), the ACME package still automatically prepends `_acme-challenge.` to both the **Domain Name** and the **DNS Alias** domain.

In the certificate entry, set:

Domain Name `company.example` which does not support automatic updates

DNS Alias Domain `dynamic.example` which is the alternative domain in a dynamic zone

DNS Domain Alias mode Leave unchecked

On the DNS server, add a CNAME record pointing to the **DNS Alias** hostname with `_acme-challenge.` prepended:

```
_acme-challenge.company.example        IN      CNAME   _acme-challenge.dynamic.example.
```

When updating, the package will update `_acme-challenge.dynamic.example` in DNS while sending `company.example` in the certificate request to the ACME provider.

Domain Alias

Domain Alias mode works similar to Challenge Alias mode but it **does not** prepend `_acme-challenge.` to the **DNS Alias** domain. Some administrators prefer this when using many hostnames in a single dynamic zone, or for working around limitations in DNS providers or platforms.

In the certificate entry, set:

Domain Name `company.example` which does not support automatic updates

DNS Alias Domain `checkme.dynamic.example` which is the alternative domain in a dynamic zone

DNS Domain Alias mode Checked

On the DNS server, add a CNAME record pointing directly to the **DNS Alias** hostname:

```
_acme-challenge.company.example        IN      CNAME   checkme.dynamic.example.
```

When updating, the package will update `checkme.dynamic.example` in DNS while sending `company.example` in the certificate request to the ACME provider.

General Settings

These settings control the general behavior of the ACME package and are not specific to any single certificate or key.

Cron Entry A checkbox which enables the ACME renewal cron job. When set, the ACME package will check all certificates each night and if any are up for renewal, it will attempt to renew them.

Write Certificates When set, the ACME package will write the certificate files out in `/conf/acme`. From there, other scripts or processes which do not support GUI integration can pick up the certificate.

29.3.4 Wildcard Certificates

Let's Encrypt supports wildcard certificates (e.g. `*.example.com`) with their ACMEv2 infrastructure. A wildcard certificate will work for any hostname inside a given domain, which helps with handling certificates for multiple domains.

Note: Unrelated to ACME, but wildcard certificates in general: A wildcard only helps for **one level** of subdomains. For example, `*.example.com` will work for `host.example.com` but will NOT work for `host.sub.example.com`. If hosts are structured in this way, a wildcard certificate is required for each sub zone, e.g. `*.sub.example.com`.

Wildcard validation **requires a DNS-based method** and works similar to validating a regular domain. For example, to get a certificate for `*.example.com`, the package updates a TXT record in DNS the same as it would for `example.com`, which means the DNS record (and potentially key name) would be for `_acme-challenge.example.com`.

To obtain a wildcard certificate, follow the same procedures as other DNS validation methods, with the following differences:

- The **Account Key** must be registered with an ACME v2 server (staging for testing, or production)
- The **Domain SAN list** should contain entries for the base domain (e.g. `example.com` and the wildcard version of the same domain (e.g. `*.example.com`. The settings will be the same for both entries.
- For *DNS-NSupdate / RFC 2136*: Set the **Key Name** to the base domain (`example.com`) for both entries.

29.4 Arping Package

arping is a utility to test the reachability and responsiveness of hosts to ARP. It is effectively like ICMP ping, except using ARP instead. This is beneficial in circumstances where the host has a firewall enabled (every host even firewalled will respond to ARP), or there is no layer 3 connectivity on the IP subnet of the host and hence cannot ping, but do have layer 2 connectivity.

The **arping** package can be very useful when trying to pick an unused IP address for a subnet to which there is not yet a route or link, but is connected at Layer 2.

See also:

Visit the arping website for more information.

29.4.1 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.5 Avahi package

The Avahi package used in pfSense® software is a system which facilitates service discovery on a local network. This means that a laptop or computer may be connected into a network and instantly be able to view other people to chat with, find printers to print to or find files being shared.

This kind of technology is already found in Apple MacOS X (branded Rendezvous, Bonjour, and sometimes Zeroconf) and is very convenient. Avahi is mainly based on Lennart Poettering's flexmdns mDNS implementation for Linux which has been discontinued in favour of Avahi.

29.5.1 Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.5.2 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.6 AWS VPC Wizard

This guide will explain how to use the AWS VPC Wizard, available in pfSense® Plus, to simplify the configuration of a VPN to a remote VPC. The administrator is asked for the minimum amount of basic information required to establish the VPN. The configurations, both on the AWS VPC side and on the pfSense® Plus side are then automatically created. When the wizard is finished executing, a functioning VPN connection to a VPC should be established.

29.6.1 AWS Access Keys

In order to connect to the AWS API to make certain required configuration changes, the AWS VPC Wizard requires Access Keys to retrieve and modify VPC configurations.

See also:

Find more information about AWS Security Credential, including Access Keys by reading AWS Security Credentials.

Access keys consist of two parts:

1. An access key ID
   - For example, **AKIAIOSFODNN7EXAMPLE**.
2. A secret access key
   - For example, **wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY**.

Access keys are like a username/password and needed for programmatic requests to AWS, including the AWS VPC Wizard. Use both the access key ID and secret access key together to authenticate requests.

Important: Manage access keys as securely as a user name and password.

Managing Access Keys

To create, modify, or delete IAM user access keys, do the following:

1. Sign in to the IAM console.
2. In the navigation bar on the upper right, choose a user name, and then choose **My Security Credentials**.
3. On the **AWS IAM Credentials** tab, in the **Access keys for CLI, SDK, and API access** section, choose **Create access key**.

4. Choose **Download .csv file** to save the access key ID and secret access key to a .csv file on your computer. Store the file in a secure location. There is no access to the secret access key again after this dialog box closes. After the .csv file has been downloaded, choose **Close**. When an access key is created, the key pair is active by default, and the pair can be used right away.
    - To disable an active access key, choose **Make inactive**.
    - To reenable an inactive access key, choose **Make active**.
    - To delete an access key, choose its **X** button at the far right of the row. Then choose **Delete** to confirm. When an access key is deleted, it's gone forever and cannot be retrieved. However, new keys can always be created.

To create, modify, or delete another IAM user's access keys, do the following:

1. Sign in to the IAM console.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys to manage, and then choose the **Security credentials** tab.
4. In the **Access keys** section, choose **Create access key**.
5. Choose **Download .csv file** to save the access key ID and secret access key to a CSV file on your computer.

Rotating Access Keys

As a security best practice, we recommend to regularly rotate (change) IAM user access keys. Rotating access keys can be done from the AWS Management Console.

To rotate access keys for an IAM user without interrupting the applications (console), create a second access key while the first access key is still active:

1. Sign in to the IAM console.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys to manage, and then choose the **Security credentials** tab.
4. In the **Access keys** section, choose **Create access key**.
5. Choose **Download .csv file** to save the access key ID and secret access key to a CSV file on your computer.
6. The new access key is active by default. At this point, the user has two active access keys.

After waiting some period of time to ensure that all applications and tools have been updated, delete the first access key:

1. Sign in to the IAM console.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys to manage, and then choose the **Security credentials** tab.
4. Locate the access key to delete and choose its **X** button at the far right of the row. Then choose **Delete** to confirm.

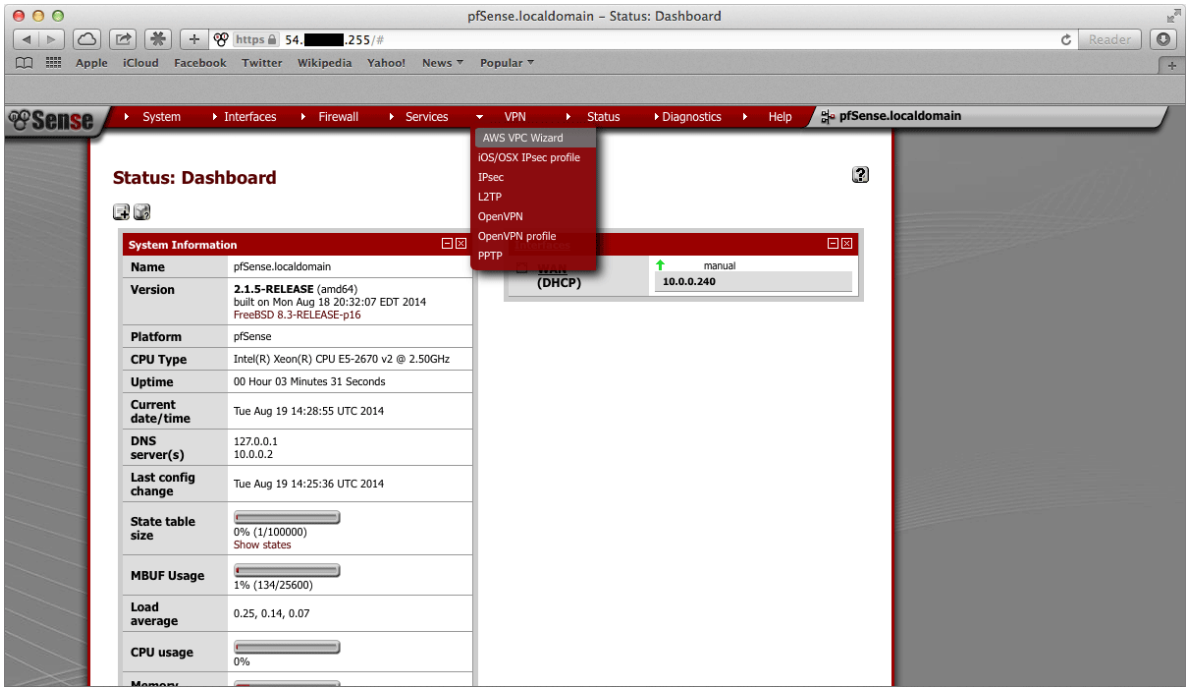
Determining When Access Keys Need Rotating

To determine when access keys need rotating (console), do the following:

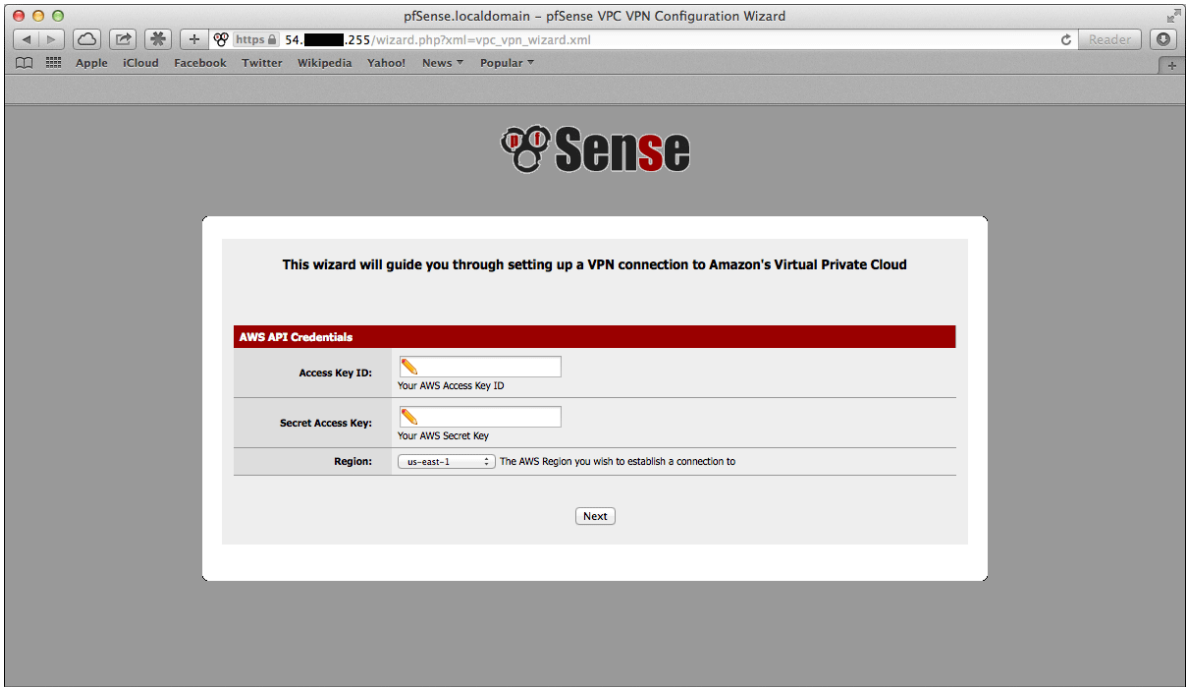
1. Sign in to the IAM console.
2. In the navigation pane, choose **Users**.
3. If necessary, add the **Access key age** column to the users table by completing the following steps:
    1. Above the table on the far right, click the settings icon.
    2. In **Manage columns**, select **Access key age**.
    3. Choose **Close** to return to the list of users.
4. The **Access key age** column shows the number of days since the oldest active access key was created. Use this information to find users with access keys that need rotating. The column displays **None** for users with no access key.

29.6.2 Using the Wizard

1. Gather your **AWS Access Key ID** and **Secret Key**.
2. Navigate to **VPN > AWS VPC Wizard** in the main menu in the Netgate® pfSense® Plus WebGUI.



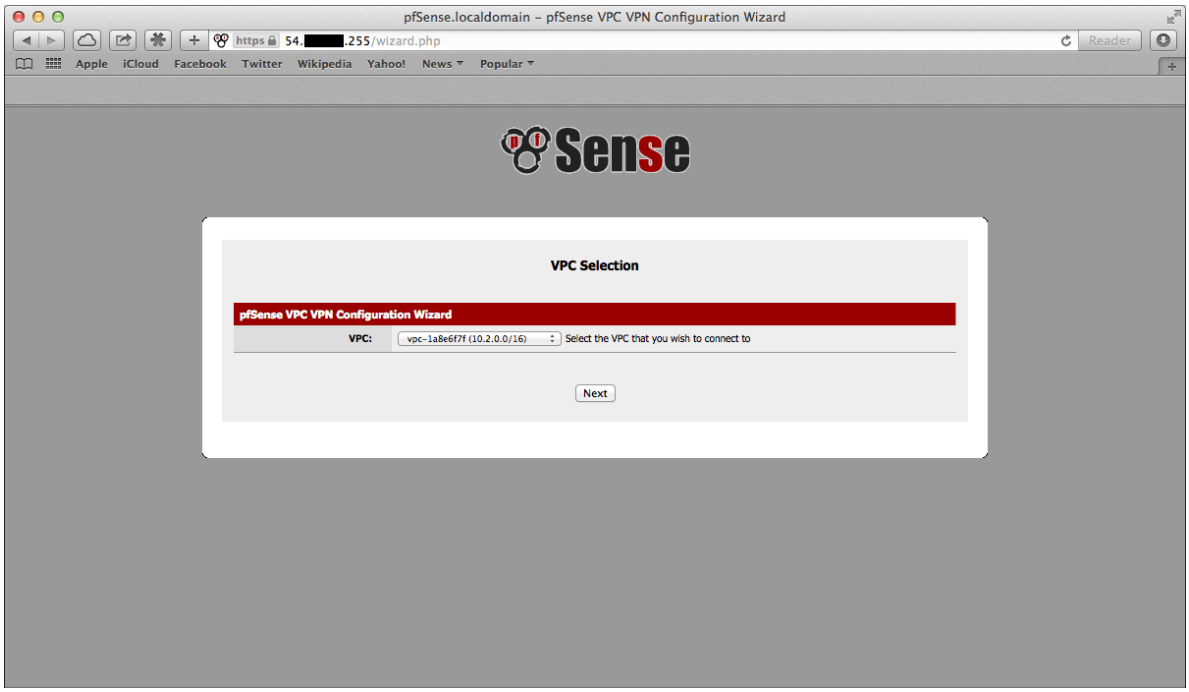
3. The first screen of the wizard prompts for the AWS Credentials. Enter the **Access Key ID** and **Secret Access Key** in the appropriate text fields and select the **Region** to connect to in the dropdown menu, then click **Next**.



The wizard will then query the AWS API using those credentials to find which VPC's exist in the selected region. If the credentials were rejected, an error message will be displayed and return to the first screen.

4. The next screen will prompt to select from the available VPC's in the selected region. Select the one to connect to from the dropdown menu. The wizard will not create a new VPC, it will only connect to an existing VPC.

   Click **Next** after selecting the desired VPC.

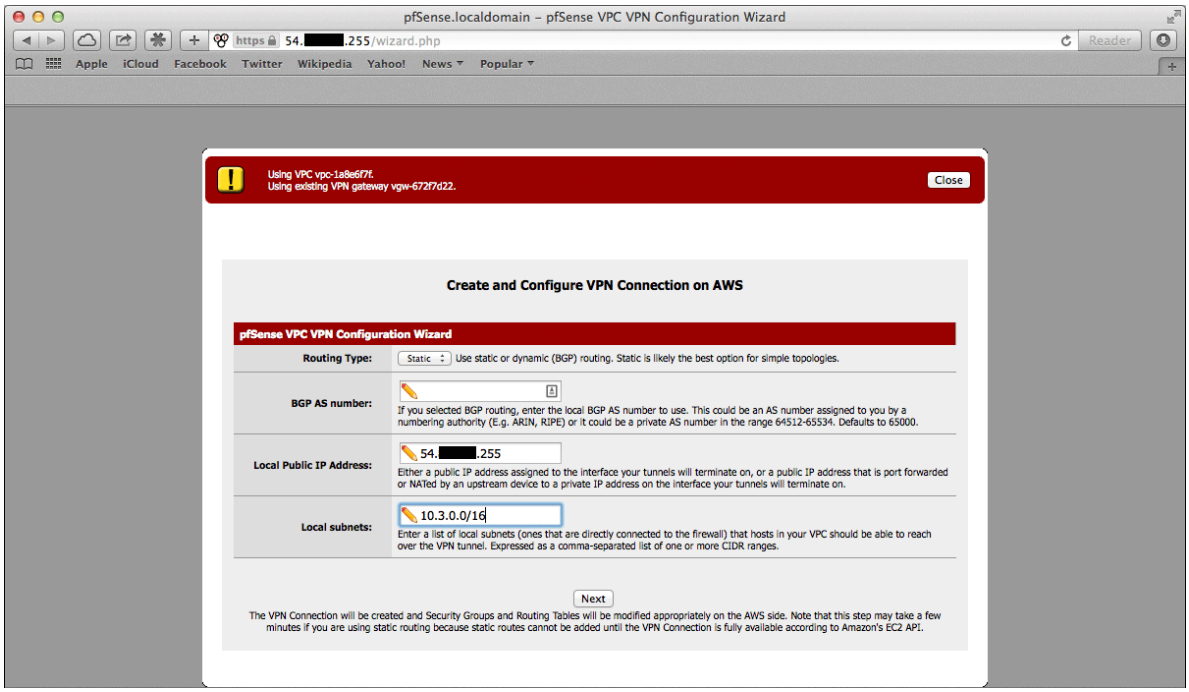


Note: If the desired VPC's to connect to isn't available, create one via the **AWS Management Console** and try

again.

The wizard will then query the AWS API to check whether there is a VPN Gateway attached to the selected VPC. If none exists, one will be created via the API. Then the next screen will be displayed.

5. On the next screen, specify routing and network data.



A description of what should be entered for each of these fields follows.

Routing Type AWS offers either **static** routing or **BGP** routing. Select the appropriate type from the dropdown menu. If unsure, static routing is likely to be adequate.

BGP AS Number If static routing was chosen, leave this field blank. Otherwise it is possible to specify an AS number to use. If left blank, the value will default to `65000`.

Local Public IP Address On an AWS Netgate appliance instance, this should be the public IP address of the Elastic IP associated with the instance. If configuring a hardware device running pfSense® Plus, this could be the public address assigned to the WAN (or other) interface of the device.

Local subnets The subnets connected to the pfSense® Plus instance that should be routed over the tunnel from hosts in the remote VPC. As an example, if connecting a pfSense® Plus instance to a remote VPC in the AWS **us-east-1** region, enter the subnets (or a single subnet) that are local to the pfSense® Plus instance and when hosts in your VPC in **us-east-1** attempted to reach addresses within those subnets, the traffic will be sent through the VPN tunnel that is being configured.

Note: When selecting static routing as the routing type, there will be a delay that is typically between 2-5 minutes before the next screen is displayed. This is because static routes must be added to the VPN Connection via the AWS API. This operation fails until the VPN Connection reaches the "available" state. This can take a few minutes to occur.

Click **Next** when done.

The wizard will then query the AWS API to find whether a Customer Gateway is configured with the selected Public IP Address. If none exists, one will be created. If one already exists, the ID will be retrieved and it will be used.

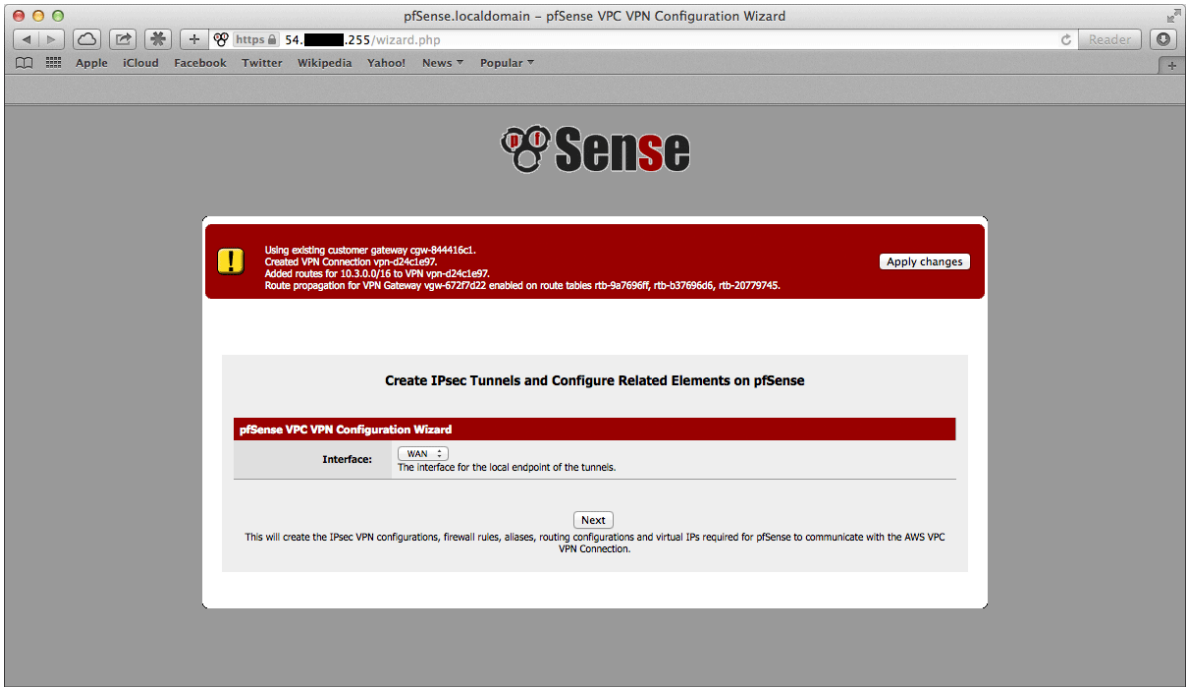
The wizard will then query the AWS API to see if a VPC Connection already exists that matches the data entered. If one exists, it will be used. Otherwise one will be created.

If static routing was selected, static routes will be added to the VPN Connection for the Local subnets entered.

Route propagation will be enabled for the VPN Gateway in each of the Route Tables that are associated with the VPC. All of these configurations are carried out in the AWS API, nothing has been changed in the pfSense® Plus VPN configurations yet.

> **Warning:** **Important Note on Billing:** Once this step is carried out and the VPN Connection is created, AWS will start billing your AWS account the hourly rate for a VPN Connection. This is $0.05 as of this writing, and that is a charge that goes entirely to AWS itself. They will do this until the VPN Connection is deleted via the **VPC Management Console**.
>
> Nothing in pfSense® Plus will ever cause AWS to stop billing for this VPN Connection. Whether it works or not, whether the pfSense® Plus instance is up or down, whether the IPsec tunnels have been deleted or reconfigured, AWS will continue to bill the hourly fee for a VPN Connection if the creation of it succeeds until it is deleted through their web interface.
>
> The wizard helps establish an initial configuration that works and configures the appropriate elements in AWS's API to facilitate this. You are responsible for making sure you understand what you're being billed for and disabling any functions, including VPN Connections, that are no longer necessary.

6. If the operations of the previous step succeeded, the next screen will appear.



Select an **Interface** to act as the local endpoint of the VPN tunnels that will be created. In most cases, this should be the WAN interface. It should generally be whatever interface is associated with the Local Public IP Address entered in the previous step.

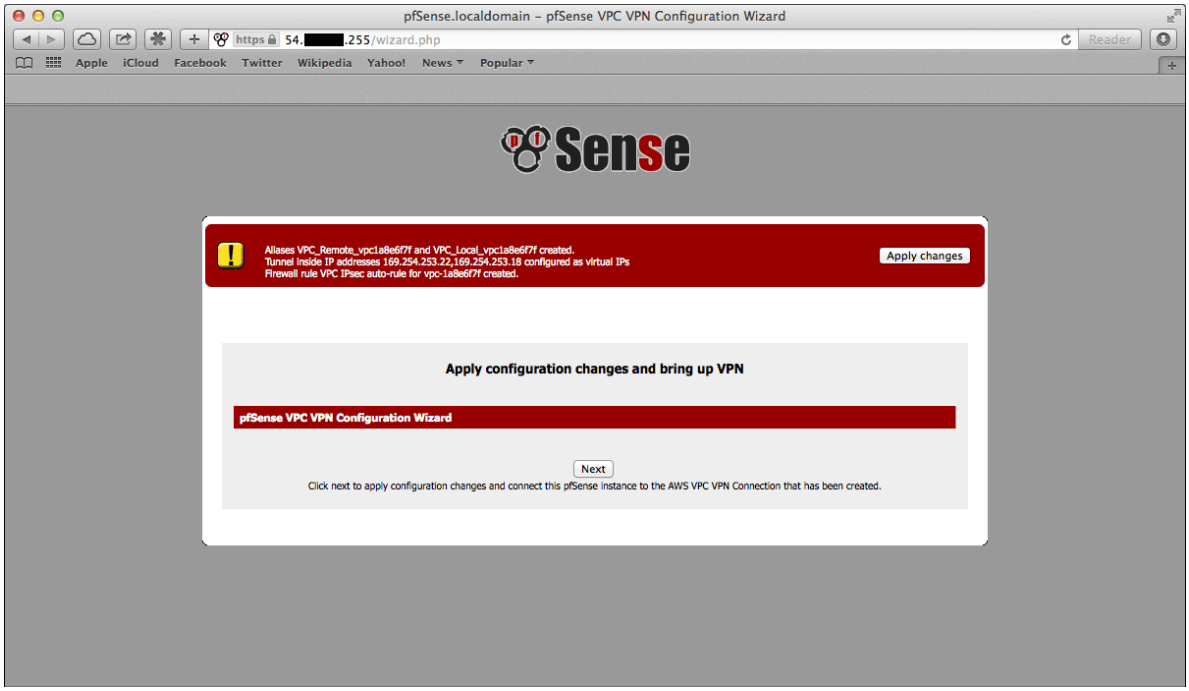
On an AWS Netgate pfSense® Plus instance, this will be whatever interface the Elastic IP is associated with.

On a hardware device running pfSense® Plus that has the Local Public IP Address directly configured on an interface, this will be the interface that the Local Public IP Address is configured on.

After clicking **Next**, the wizard will configure the VPN and associated settings within pfSense® Plus itself using data returned by the AWS API in the previous step.

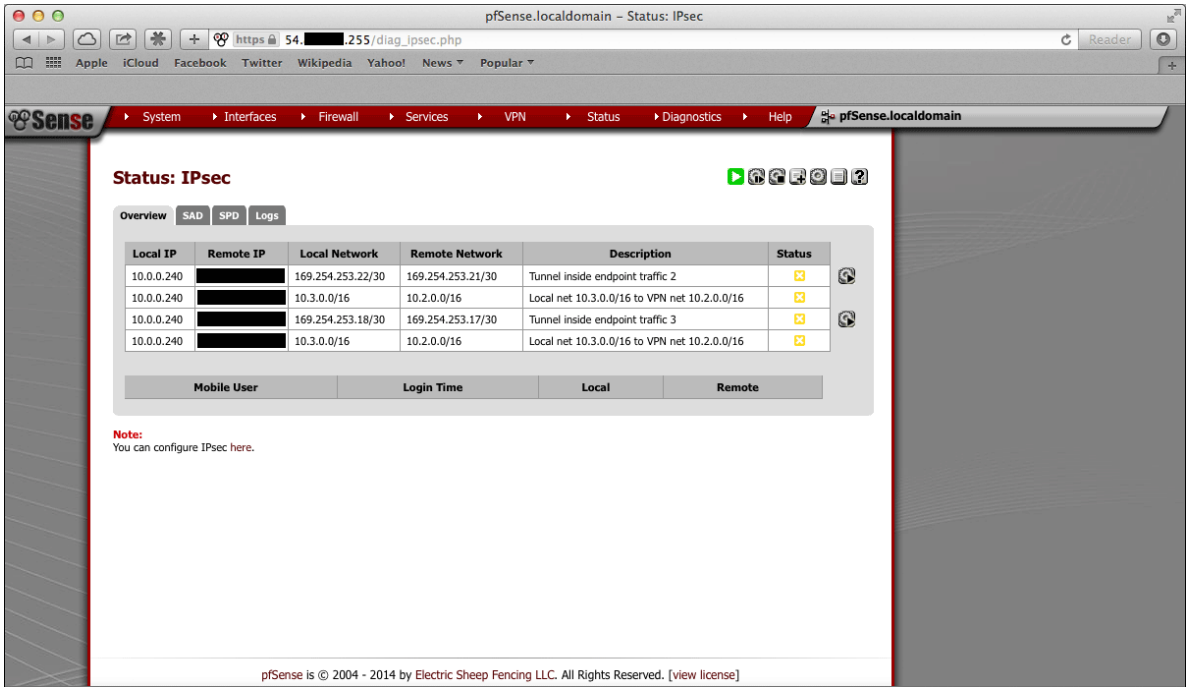
It will configure 2 IPsec tunnels, a firewall rule, 2 Aliases (referenced by the firewall rule), and 2 Virtual IP Addresses. If **BGP** was selected as the **Routing Type** in the previous step, it will install the **FRR** package automatically and configure it appropriately.

7. The next screen will appear and prompt to apply the configuration changes that have been made.



After clicking **Next**, all the configuration changes that were made will be applied.

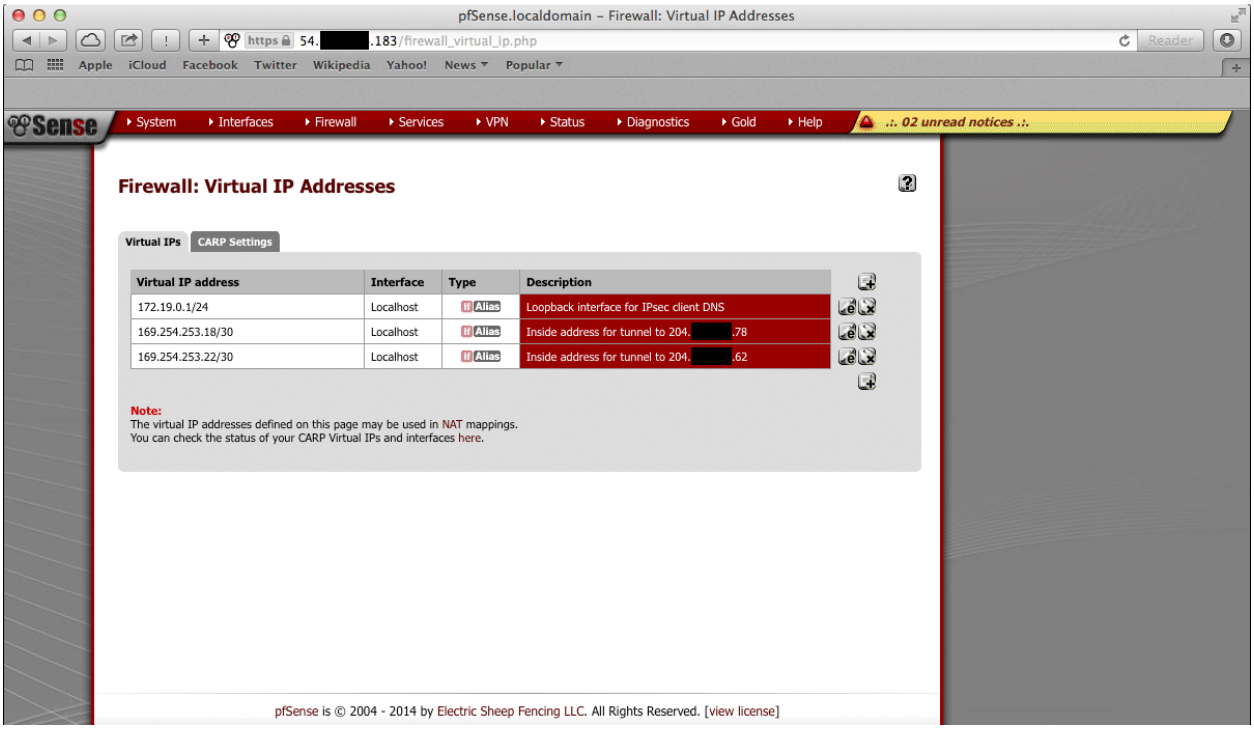
8. The wizard will be completed and the browser will be redirected to the IPsec status page. The VPN to the VPC should now be fully configured.



Note: Sometimes there is a delay of 5-10 minutes before the tunnels are fully functional and passing traffic. This has been observed particularly often during the setup of tunnels using BGP routing.

29.6.3 Testing Connectivity

Verify that the IPsec tunnels are functioning by attempting to ping the "inside tunnel addresses" of the VPC side of the tunnel by navigating to **Firewall -> Virtual IP**. There should be two virtual IP addresses configured that have Descriptions like "Inside address for tunnel to <remote IP address>".



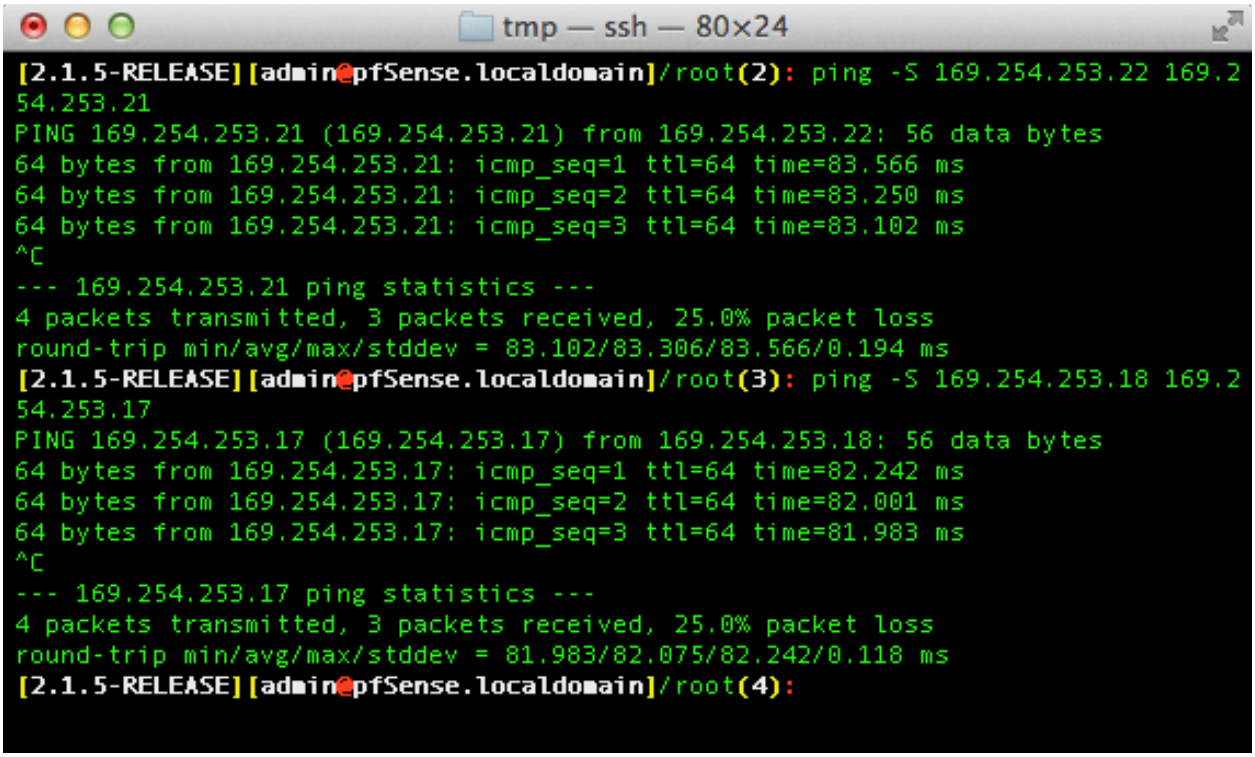
Amazon provides inside addresses for each end of the tunnel in a /30 subnet in IPv4 link local address space (169.254.x.y). Typically, the first usable address in the /30 is the inside address for the VPC end of the tunnel and the other usable address is the inside address for your end of the tunnel. So if you ping from one of the virtual IP addresses configured on the pfSense® Plus instance to the IP address that is one less (for example, if the virtual IP address were **169.254.253.22**, ping from **169.254.253.22** to **169.254.253.21**), that checks whether the other end of the tunnel is responding and whether the tunnel is functioning properly.

Note: It sometimes takes a few minutes for the tunnels to begin working after the configuration wizard completes.

To ping, log into the pfSense® Plus instance via SSH and execute `ping` from a shell prompt. For the previous example of **169.254.253.22**, the proper syntax is:

```
ping -S 169.254.253.22 169.254.253.21
```

When pinging from a shell prompt, it is possible leave the command running indefinitely and interrupt it there is a response.



It is also possible to ping via the **Diagnostics -> Ping** page by selecting the appropriate source address and entering the remote tunnel inside address in the **Host** field. This will only send a limited number of ping packets, so it may be necessary to repeat this a few times.

29.6.4 VPC Configuration Details

The AWS documentation for connecting a hardware device to a VPC provides a great amount of detail on configuring VPN Connections to VPC.

The main configuration elements that exist in Amazon's data model are the Virtual Private Gateway, the Customer Gateway, and the VPN Connection. In order to configure a VPN connection to a VPC, all 3 of these need to exist. The Virtual Private Gateway is the VPN endpoint on Amazon's side. The customer gateway is the VPN endpoint on the Netgate® pfSense® Plus instance/device being configured. The VPN Connection is the IPsec VPN between Amazon and the pfSense® Plus instance/device.

A Virtual Private Gateway needs to exist and be associated to the VPC. A particular VPN Gateway can only be associated to one VPC at a time. Once the VPC to connect to has been selected, the wizard invokes the AWS API call **DescribeVpnGateways** to determine if a VPN Gateway already exists that is attached to the VPC. If none exists, it creates one with the **CreateVpnGateway** API call and attaches it to the VPC with the **AttachVpnGateway** call.

A Customer Gateway needs to be created for the public IP address of the device or virtual machine that will be used to connect to the VPC. The wizard invokes the AWS API call **DescribeCustomerGateways** to determine if a Customer Gateway already exists. If none exists, one is created with the **CreateCustomerGateway** API call.

The VPN Connection connects the Virtual Private Gateway and Customer Gateway. The wizard checks to see if there is an existing VPN Connection configured that connects those endpoints by invoking the AWS API call **DescribeVpnConnections**. If none exists, one is created using the **CreateVpnConnection** call. One of the fields returned by this call is a block of XML configuration data that contains configuration data assigned by AWS for use with configuring the VPN Connection. This data is stored and used in subsequent steps to make the required configuration changes within pfSense® Plus.

Any objects created through API calls by the wizard will be tagged with names like `auto-created by pfSense VPC <your_VPC_ID>`. This is accomplished by calling the AWS API call **CreateTags** and using `Name` as the key for the tag.

In addition to creation of the items mentioned above, required adjustments are made to Security Groups and Route Tables to facilitate communication over the VPN. The Security Groups associated with the VPC are updated to allow inbound access from the local subnets on the pfSense® Plus end of the VPN. They are checked first via the **DescribeSecurityGroups** AWS API call to determine if the access is already allowed. Any of the subnets that is not already allowed has inbound access added via the **AuthorizeSecurityGroupIngress** AWS API call.

Route Tables associated with the VPC are updated to receive routes from the VPN Gateway used by the VPN Connection. They are checked first via the **DescribeRouteTables** AWS API call. If the VPN Gateway ID is not included in the list of VPN Gateways propagating routes, route propagation for the VPN Gateway is enabled on that table using the **EnableVgwRoutePropagation** AWS API call.

For VPN Connections using static routing, static routes for the specified subnets are added to the VPN Connection. This is done via the **CreateVpnConnectionRoute** AWS API call.

29.6.5 pfSense Plus Configuration Details

On the pfSense® Plus side, there are numerous configurations added to support the VPN to the VPC.

Aliases

First, aliases are created for use in a firewall rule. These aliases are intended to contain the subnets that traffic should be allowed to ingress over the IPsec tunnel. One alias represents the local subnets on the pfSense® Plus side and is given a name like **VPC_Local_vpc_12345678** and the other represents the remote subnets on the VPC side and is given a name like **VPC_Remote_vpc_12345678**.

Virtual IP addresses

Next, virtual IP addresses are added on the **lo0** (loopback) interface. These virtual IP addresses are the local "inside addresses" of the IPsec tunnels. These addresses are used as the local address for BGP communication when BGP routing is selected. These addresses are IPv4 link local addresses (see RFC 3297). AWS assigns /30's out of the network 169.254.0.0/16 for this purpose.

Note: These addresses are also useful as a ping target to execute a basic test of whether the tunnel is functioning properly. Executing a ping from a source address of one of these IP addresses to the corresponding inside address of the other end of the tunnel helps determine whether the tunnel negotiation is completing properly.

Firewall rules

Next, a firewall rule is added on the IPsec interface that allows traffic from the VPC networks to the local subnets. This rule uses the previously created Aliases as source/destination targets.

IPsec

Then, IPsec phase 1 and phase 2 associations are set up. Most of the settings required are extracted from a block of XML data that was returned by the **CreateVpnConnection** call made during the AWS configuration step. This includes parameters like endpoint IP addresses, encryption ciphers, timer values, etc.

If BGP routing was selected, the configurations for the FRR BGP daemon are established. The required settings are determined using the AS number entered into the wizard and the parameters returned by the **CreateVpnConnection** call made during the AWS configuration step.

29.6.6 Tested Configurations

Various topologies may be possible to establish using the AWS VPC Configuration Wizard. This section enumerates some of the configurations that were successfully tested.

Platforms

- Various hardware platforms.
- 64-bit virtual machines in VMware vSphere/ESX.
- Amazon EC2 instances (Xen virtual machines) of the Netgate® pfSense® Plus Router/Firewall/VPN AWS AMI.

Local network/routing configurations

- pfSense® Plus with a public address configured on the WAN interface.
- pfSense® Plus with a private address configured on the WAN interface behind a 1:1 NAT.
- pfSense® Plus with a private address configured on the WAN interface behind a PAT (1:many NAT).

VPC Topologies

- pfSense® Plus connected to a single VPC.
- pfSense® Plus connected to multiple VPC's in different regions.
- Amazon EC2 instance of the Netgate pfSense® Plus Router/Firewall/VPN AWS AMI connected to a VPC belonging to the same AWS account in a different region.
- Amazon EC2 instance of the Netgate pfSense® Plus Router/Firewall/VPN AWS AMI connected to a VPC belonging to a different AWS account in the same region.

The configuration recommended for the greatest amount of stability is to have a public IPv4 address directly configured on the WAN interface of your Netgate pfSense® Plus firewall, but VPNs have been successfully established under all of the conditions listed above.

Whether any of these solutions is appropriate should be evaluated in the context of personal needs and existing infrastructure. Other configurations not listed above may be possible as well.

29.6.7 AWS VPC Wizard FAQ

1. What level of redundancy is provided by the two tunnels?

Amazon provides two tunnel endpoints that will allow traffic to be sent between your networks and the remote VPC you are connected to. The racoon daemon in pfSense® Plus is only capable of establishing an active phase 2 association for a particular source/destination pair on a single tunnel.

Phase 2 associations between the local subnets and the remote VPC subnet are configured in the pfSense® Plus WebGUI for both tunnels, but racoon will only actually establish an association for the first tunnel. This means that racoon will only ever try to send traffic destined for the remote VPC subnet over the first tunnel.

If that tunnel goes down, the second tunnel may be up and inbound traffic from the remote VPC may be sent to your local networks over that tunnel automatically. But outbound traffic to your remote VPC would not automatically fail over to the second tunnel. In order for you to send your outbound traffic over the second tunnel, you would need to disable the phase 2 associations for the first tunnel and apply the changes.

2. I quit the wizard before finishing. Now what?

To finish setting up the VPN, go back to the wizard and run through it again. It should reuse any partial configurations that were generated before it was stopped and create the new elements that are required.

3. What are the AWS charges for this?

AWS determines their own pricing and provides details for EC2 pricing and VPC pricing. There are many types of charges that may be incurred for operating instances on AWS (e.g. charges related to running an instance, bandwidth, storage, elastic IPs, etc).

The charge of specific interest in this case is the hourly charge for a VPN Connection. As of this writing, it costs $0.05 (USD) per hour in most regions to have a VPN Connection configured and available. AWS will charge whether the VPN Connection is being used or not as long as it is configured. This will be configured by the third step of the wizard and will never be removed by pfSense® Plus.

If the VPN Connection is no longer needed and billing for it needs to be stopped, visit AWS's **VPC Management Console** and delete the VPN Connection manually.

4. Can I use the wizard to connect to the GovCloud region?

This hasn't been officially tested, but at least one user has reported that they were able to successfully connect to the GovCloud region. They manually added the region **us-gov-west-1** to the list of regions in the first step of the wizard and were able to successfully connect to their VPC in that region. This may be supported in a future build, but to try without official support, do the following:

1. Under the **System > Advanced** menu, make sure the **Enable Secure Shell** box is checked. This is already done by default on AWS instances, but is off by default on Netgate hardware devices with pfSense® Plus software.
2. Log into the instance via SSH.
3. Make sure the root filesystem is mounted as read/write. On an AWS instance or a hardware device running on an SSD, this should be true. On a hardware device using Compact Flash or an SD card for storage, it will probably be necessary to remount the root filesystem in read/write mode by running:

```
mount -uw /
```

4. Edit the file **/usr/local/www/wizards/vpc_vpn_wizard.xml** using `vi`. Look for a section of the file that looks like this:

```
<option>
    <name>sa-east-1</name>
    <value>sa-east-1</value>
</option>
```

That should appear directly after several similar **<option>** specifications containing all of the other available regions. Right underneath that section, add the following:

```
<option>
    <name>us-gov-west-1</name>
    <value>us-gov-west-1</value>
</option>
```

Then save the file and exit vi.

5. If the filesystem had to be remounted in read/write mode earlier, remount it in read-only mode by running:

The GovCloud region should now appear as a choice in the first step of the wizard.

29.7 Backup Files and Directories with the Backup Package

The **Backup** package allows any given set of files/folders on the system to be backed up and restored. For most, this is not necessary, but it can be useful for backing up RRD data or for packages that may have customized files that are not kept in config.xml.

To install the package:

- Navigate to **System > Packages**
- Locate **Backup** in the list
- Click **Install** at the end of its entry
- Click **Confirm** to begin the installation

Once installed, the package is available at **Diagnostics > Backup Files/Dir**. It is fairly simple to use, as shown in the following example.

29.7.1 Backing up RRD Data

Using this Backup package it is quite easy to make a backup of RRD graph data outside of the config.xml method.

See also:

Monitoring Graphs

- Navigate to **Diagnostics > Backup Files/Dir**
- Click **Add** to add a new location to the backup set
- Enter `RRD Files` in the **Name** field
- Enter `/var/db/rrd` in the **Path** field
- Set **Enabled** to *True*

- Enter `RRD Graph Data Files` in the **Description**
- Click **Save**
- Click the **Backup** button to download the backup archive, which contains the configured files and directories for the backup set.
- Save the file in a safe location and consider keeping multiple copies if the data is important.

29.7.2 Restoring RRD Data

- Navigate to **Diagnostics > Backup Files/Dir**
- Click **Browse**
- Locate and select the backup archive file downloaded previously
- Click **Upload** to restore the files

For this example, because the RRD files are only touched when updated once every 60 seconds, it is not necessary to reboot or restart any services once the files are restored.

29.8 Cache / Proxy

Proxies are intermediaries that sit between clients and servers. A client connects to a proxy, and then the proxy decides if the client can receive content from a server. If so, the proxy makes its own connection to the server and then passes back data to the client.

There are two major types of proxies:

Forward Proxy Typically sits between local clients and remote Internet servers. It can be used to control which web sites that clients are allowed to load, or log servers and URLs clients are visiting. These mostly work with HTTP, but in special cases can also work with HTTPS.

Reverse Proxy Typically sits between remote clients and local servers. These allow for load balancing, failover, or other intelligent connection routing for public services such as web servers.

29.8.1 Squid

Squid is primarily a forward proxy used for client access control. It can, however, be used in a reverse proxy role if needed. The reverse proxy capabilities are inferior to HAProxy, however.

The most common use case for squid is covered in *Configuring the Squid Package as a Transparent HTTP Proxy*. Additional documentation below covers related topics.

Tuning the Squid Package

Performance Tweaks

Some users have reported that making the following change has greatly increased performance:

1. Edit **/boot/loader.conf.local**
2. Change `kern.ipc.nmbclusters="0"` to `kern.ipc.nmbclusters="32768"`
3. Reboot the pfSense® router

See *Hardware Tuning and Troubleshooting* for more information on that setting.

Some people have also seen better performance by using the *ufs* cache filesystem setting. When using *ufs* filesystem, **vfs.read_max=32** may be increased to **vfs.read_max=128** in **System > Advanced**, **Sytem Tunables** tab.

Compact swap.state

Squid keeps a cache index journal called **swap.state** in the top level of the squid cache folder, typically **/var/squid/cache/swap.sate**. **This file can grow very large and consume all hard drive space**. To ensure this does not happen, set a **Log Rotate** value in the squid configuration.

By setting a number of days to retain the logs, the squid package will activate a nightly cron job which runs:

```
squid -k rotate
```

Part of this rotation process includes compacting the swap.state file, keeping it from getting too large.

If this file is too large and needs to be removed, this may be done while squid is running. After the file is removed, run:

```
squid -k rotate
```

This will cause it to be written out again (but compacted). Alternately, tell squid to perform a clean shutdown with:

```
squid -k shutdown
```

This will also write the **swap.state** file out again, but squid will stop after this and must be restarted, so it is a less desirable option.

If the **swap.state** file is removed while squid is *not* running, it will have to completely rescan the cache folder to rebuild it once squid is restart. This can be a lengthy and time consuming process. It may be better to remove the contents of the existing cache folder, and rebuild the structure again by running:

```
squid -z
```

See the Squid FAQ entry for more details.

Random Tips/Tricks

Warning: There could be any number of reasons not to do the following things. Be careful, and test any changes.

Caching Windows Updates

Warning: These settings could break things, but when it works it works beautifully!

If there are a bunch of local PCs that need Windows Updates, but a WSUS server is not an option, squid can cache them.

- On the **General Settings** tab of the squid configuration (**Services > Proxy Server**), place the following patterns recommended by Squid in the **Custom Options** box at the bottom:

```
refresh_pattern -i windowsupdate.com/.*\.
↪(cab|exe|ms[i|u|f|p]|[ap]sf|wm[v|a]|dat|zip|psf) 43200 80% 129600 reload-into-
↪ims
refresh_pattern -i microsoft.com/.*\.
↪(cab|exe|ms[i|u|f|p]|[ap]sf|wm[v|a]|dat|zip|psf) 43200 80% 129600 reload-into-
↪ims
refresh_pattern -i windows.com/.*\.
↪(cab|exe|ms[i|u|f|p]|[ap]sf|wm[v|a]|dat|zip|psf) 43200 80% 129600 reload-into-
↪ims
refresh_pattern -i microsoft.com.akadns.net/.*\.
↪(cab|exe|ms[i|u|f|p]|[ap]sf|wm[v|a]|dat|zip|psf) 43200 80% 129600 reload-into-
↪ims
refresh_pattern -i deploy.akamaitechnologies.com/.*\.
↪(cab|exe|ms[i|u|f|p]|[ap]sf|wm[v|a]|dat|zip|psf) 43200 80% 129600 reload-into-
↪ims
range_offset_limit none
```

- Click **Save**
- On the **Cache Management** tab of the Squid configuration:
  - Change **Hard Disk Cache size** to something large, say *3000* or *4000* (3GB or 4GB), to accommodate the updates.
  - Change the **Maximum object size** to something big, such as *512000* for 512MB. Going bigger may be needed if any updates larger than that size are released.
- Click **Save**

Caching Mac Updates

```
refresh_pattern ([^.]+.|)(download|adcdownload).(apple.|)com/.*\.(pkg|dmg) 4320 100%␣
↪43200 reload-into-ims
```

Caching AVG and other Updates

As above, but add this before the other options. The same warnings apply:

```
refresh_pattern ([^.]+.|)avg.com/.*\.(bin) 4320 100% 43200 reload-into-ims
refresh_pattern ([^.]+.|)spywareblaster.net/.*\.(dtb) 4320 100% 64800 reload-into-ims
refresh_pattern ([^.]+.|)symantecliveupdate.com/.*\.(zip|exe) 43200 100% 43200 reload-
↪into-ims
refresh_pattern ([^.]+.|)avast.com/.*\.(vpu|vpaa) 4320 100% 43200 reload-into-ims
```

Tweaking Update Caching / Squid seems to download on its own

Change This:

```
range_offset_limit none
```

To:

```
range_offset_limit 0
```

As an alternative, also try:

```
quick_abort_min 0 KB
quick_abort_max 0 KB
```

Or:

```
quick_abort_pct 70
```

To ensure that a file is only downloaded if a user actually receives 70% or more of it. Otherwise if a user requests a file and then aborts, it will download the whole file.

Note: If **range_offset_limit** is set to `-1` the quick abort options will NOT work

Parent proxy

Setting parent proxy available at the **Proxy server: Upstream proxy settings** tab. In most cases, these settings work if the parent proxy also squid.

To use a parent proxy on another server (not squid), it is necessary to disable **Upstream proxy settings**, and use the **Custom options** in the **Proxy server: General settings** tab. For example for use as a parent proxy installed package HAVP need to add the line *cache_peer 127.0.0.1 parent 3121 7 no-query* (Here HAVP configured to use the IP address *127.0.0.1* port *3121*).

Configuring the SquidGuard Package

squidGuard is a URL redirector used to integrate blacklists with the Squid proxy software. There are two big advantages to squidGuard: it is fast and it is free. squidGuard is published under the GNU Public License.

squidGuard can be used to:

- Limit the web access for some users to a list of accepted/well known web servers and/or URLs only.
- Block access to some listed or blacklisted web servers and/or URLs for some users.
- Block access to URLs matching a list of regular expressions or words for some users.
- Enforce the use of domain names/prohibit the use of IP addresses in URLs.
- Redirect blocked URLs to an info page.
- Redirect banners to an empty GIF.
- Have different access rules based on time of day, day of the week, date etc.

Installing Squid and squidGuard

1. From the pfSense® webGUI, navigate to **System > Packages**, **Available Packages** tab
2. Install the **Squid** package if it is not already installed.
3. Install the **squidGuard** package
4. Configure **Squid** package.
5. Configure **squidGuard** package.

Configure the squidGuard Package

Basic configuration

Here describes how to enable and configure squidGuard, and common users access.

1. Open **General settings** tab.
    1. Check the **Enable** box to activate the package.
    2. Set **Blacklist** options to use blacklist categories. (See above, optional)
    3. Click **Save** button.
2. Open **Common ACL** page.
    1. Click **Target Rules List** to show defined blacklists and target categories
        1. Define default user access: select **Default access [all]** as *allow* or *deny*.
        2. Define other category actions:
            1. Select **—**, to ignore a category.
            2. Select **allow**, to allow this category for clients.
            3. Select **deny**, to deny this category for clients.
            4. Select **white**, to allow this category without any restrictions. This option is used for exceptions to prohibited categories.
        3. To prohibit clients from using IP addresses in URLs, check **Do Not Allow IP Addresses in URL**.
        4. Select **Redirect mode**:
            1. *Int error page*: Use the built-in error page. A custom message may be entered in the **Redirect info** box below.
            2. *Int blank page*: Redirect to a blank page
            3. The other options are various redirects to external error pages, and a URL must be entered in the **Redirect info** box if they are chosen.
        5. **Use safe search engine**: Protect customers from unwanted search results. It is supported by *Google, Yandex, Yahoo, MSN, Live Search*. Make sure that these search engines are available. If this protection should be strictly enforced, disable access to all other search engines.
3. After settings are complete, return to the **General Settings** tab and press **Apply**.

Blacklist

Blacklists are optional, but often useful for allowing access to certain types of sites.

squidGuard comes with a small blacklist basically for testing purposes. They should not be used in production. A better way is to start with one of the blacklist collections listed (alphabetically) below.

- MESD blacklists - They are freely available.
- Shalla's Blacklists - Free for non commercial/private use. (Recommended)
- more..

Downloading blacklists:

1. Open **General Settings** tab in squidGuard package GUI, found at **Services > Proxy Filter**.
2. Check **Blacklist** to enable the use of blacklists.
3. Enter blacklist URL in the field **Blacklist URL**.
4. If the firewall is itself behind a proxy, enter the proxy information in **Blacklist proxy** (this step is not necessary for most people).
5. Click **Save**.
6. Navigate to the **Blacklist** tab inside of squidGuard.
7. Click the **Download** button.
8. Wait while blacklist will downloaded and prepared to use (10-35 min). Progress will be displayed on that page as the list is downloaded and processed.

How-Tos

Exclude domain/URL from blacklist

In the squidGuard GUI (**Services > Proxy Filter**):

1. Open the **Target categories** page
2. Click  to add a new item
3. Enter a name for the category - `myWhitelist` for example.
4. Add domains and/or URLs to the lists as needed. Entries should be separated by a space. The examples on the page show how entries should be formatted.
5. As with the Common ACL discussed previously, redirect and logging options specific to this category may be set.
6. Click **Save**.
7. Open **Common ACL** or **Groups ACL** page (whichever should have an exclusion).
8. Click **Target Rule List** to expand the list of categories. The newly created category should show alphabetically in the list, above any blacklist categories. Find the **MyWhiteList** entry in the list and select **white**.
9. Click **Save**.
10. Return to the **General Settings** tab and press **Apply**.

Block download by Extension

In the squidGuard GUI (**Services > Proxy Filter**):

1. Open the **Target categories** page.
2. Click  to add a new item.
3. Enter a name for the category - `myBlockExt` for example.
4. Add Expressions (for example for asf, zip, exe and etc files):

```
(.*\/.*\.(asf|wm|wma|wmv|zip|rar|cab|mp3|avi|mpg|swf|exe|mpeg|mp.|mpv|mp3|wm.
↪|vpu))
```

5. Click **Save**.
6. Open **Common ACL** or **Groups ACL** page (whichever should have an exclusion).
7. Click **Target Rule List** to expand the list of categories. The newly created category should show alphabetically in the list, above any blacklist categories. Find the **myBlockExt** entry in the list and select **deny**.
8. Click **Save**.
9. Return to the **General Settings** tab and press **Apply**.

Troubleshooting

Netflix

If Netflix will not load while squidGuard is active, it is likely because Netflix requires accessing URLs by IP address. Ensure that ACLs matching clients allowed to reach Netflix also have **Do not allow IP-Addresses in URL** unchecked.

Service Does Not Start

If the squidGuard service will not start, there are a few possible explanations:

- On all versions of Squid, if **only** blacklists have been configured, then at startup some important files/directories may not be set properly.
  - Add at least one **Custom Target Category** with a site to pass or block and use it along with the blacklist entries to work around the problem.
- On squid 3.x, the squidGuard service will only start when traffic requires it to run, so it can appear to be stopped even when working properly.
  - Only worry about the service if it appears to not work, don't count on the service status alone.

Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

See also:

- *Authenticating Squid Package Users with FreeRADIUS*
- *Setting up WPAD Autoconfigure for the Squid Package*
- *Troubleshooting the Squid Package*

29.8.2 HAProxy

HAProxy is a powerful reverse proxy that can handle many different types of tasks and scales well for large deployments.

- *HAProxy package*
- *Troubleshooting the HAProxy Package*

29.9 FreeRADIUS package

FreeRADIUS is a free implementation of the RADIUS protocol. Supports MySQL, PostgreSQL, LDAP, Kerberos.

Refer to the following articles for more information on the listed topics:

29.9.1 Testing the FreeRADIUS Package

Testing the *FreeRADIUS Package* on a pfSense® firewall.

Test Configuration

At a minimum, testing FreeRADIUS requires A **User**, an **Interface**, and a **NAS/Client**.

- Add a **User** with the following configuration:

  Username `testuser`

  Password `testpassword`

- Add a **Client/NAS** with the following configuration:

  IP Address `127.0.0.1`

  Shared Secret `testing123`

- Add an **Interface** with the following configuration:

  IP Address `127.0.0.1`

Interface Type *Auth*

Port `1812`

GUI Test

The easiest way to test is by using **Diagnostics > Authentication** in the GUI.

First, add a RADIUS server entry to the user manager as described in *Authentication Servers*.

- Navigate to **System > User Manager**, **Authentication Servers** tab
- Fill in the settings to match the entry in FreeRADIUS:

  Descriptive Name `FreeRADIUS`

  Type *RADIUS*

  Hostname or IP Address `127.0.0.1`

  Shared Secret `testing123`

  Services Offered *Authentication*

  Authentication Port `1812`

- Click **Save**

Next, perform the GUI test:

- Navigate to **Diagnostics > Authentication**
- Set **Authentication Server** to the RADIUS server in the user manager
- Fill in the **Username** and **Password**
- Click  **Test**

If the test succeeds, the GUI prints a success message:

> User testuser authenticated successfully.

The system log will also contain a message indicating a successful login:

```
radiusd[44793]: Login OK: [testuser/testpassword] (from client testing port 0)
```

If the test fails, the GUI prints a failure message:

> Authentication failed.

The system log will also contain a message indicating failure:

```
radiusd[44793]: Login incorrect: [testser/testpassword] (from client testing port 0)
```

CLI Test

FreeRADIUS offers an easy to use command line tool to check if the server is running and listening to incoming requests.

SSH to the firewall, start a shell, and type in the following command:

```
radtest testuser testpassword 127.0.0.1:1812 0 testing123
```

The following output will appear if the test succeeds:

```
: radtest testuser testpassword 127.0.0.1:1812 10 testing123
Sending Access-Request of id 1 to 127.0.0.1 port 1812
       User-Name = "testuser"
       User-Password = "testpassword"
       NAS-IP-Address = 192.168.0.22
       NAS-Port = 10
       Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=1, length=20
```

The `Access-Accept` portion of the output is the most relevant.

Check the system log for the following output:

```
radiusd[44793]: Login OK: [testuser/testpassword] (from client testing port 10)
```

If a part of the test fails, such as incorrect username, then the test command output will look like the following:

```
: radtest testser testpassword 127.0.0.1:1812 10 testing123
Sending Access-Request of id 104 to 127.0.0.1 port 1812
       User-Name = "testser"
       User-Password = "testpassword"
       NAS-IP-Address = 192.168.0.22
       NAS-Port = 10
       Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Reject packet from host 127.0.0.1 port 1812, id=104, length=20
```

The `Accesss-Reject` packet indicates that the server rejected the attempt, and the system log will contain the following output:

```
radiusd[44793]: Login incorrect: [testser/testpassword] (from client testing port 10)
```

See also:

- *Using Mobile One-Time Passwords with FreeRADIUS*
- *Using EAP and PEAP with FreeRADIUS*
- *Zone Configuration Options*
- *Authenticating OpenVPN Users with FreeRADIUS*
- *Authenticating Squid Package Users with FreeRADIUS*

29.9.2 Features

The features below were tested on pfSense software version 2.x

- Authentication with Captive-Portal
- Pre-defined user attributes and custom check-items and reply-items
- NAS/Clients running on IPv4 and IPv6
- Interfaces can listen on IPv4 and IPv6
- OpenVPN + Username + RADIUS and OpenVPN + Username + Cert + RADIUS
- Auth with PAP, CHAP, MSCHAP, MSCHAPv2
- Auth with EAP-MD5 + dynamic VLAN assignment
- Auth with PEAP + dynamic VLAN assignment
- Auth with EAP-TLS/EAP-TTLS + dynamic VLAN assignment

```
radiusd[3206]: Login OK: [testuser/<via Auth-Type = EAP>] (from client pfsense␣
↪port 0 cli 00-04-23-5C-9D-19)
radiusd[3206]: Login OK: [testuser/<via Auth-Type = EAP>] (from client pfsense␣
↪port 0 cli 00-04-23-5C-9D-19)
radiusd[3206]: Login OK: [testuser/<via Auth-Type = EAP>] (from client pfsense␣
↪port 0 via TLS tunnel)
radiusd[3206]: Login OK: [testuser/<via Auth-Type = EAP>] (from client pfsense␣
↪port 0 via TLS tunnel)
```

- Simultaneous-Use - The following will be present in the system log

```
radiusd[3206]: Multiple logins (max 1) : [testuser/testpw] (from client testing␣
↪port 10)
```

- A certain amount of time per day/week/month/forever (`CHECK-ITEM: Max-Daily-Session := 60`) The user will be disconnected and cannot re-login after the amount of time is reached:

```
radiusd[3206]: Invalid user (rlm_counter: Maximum daily usage time reached):␣
↪[testuser/<via Auth-Type = EAP>] (from client pfsense port 0 cli 00-04-23-5C-9D-
↪19)
```

- A certain amount of traffic per day/week/month/forever. The user will be disconnected and cannot re-login after the amount of traffic is reached. The syslog output looks like this:

```
root: FreeRADIUS: Used amount of daily upload and download traffic by testuser is␣
↪0 of 100 MB! The user was accepted!!!
root: FreeRADIUS: Credentials are probably correct but the user testuser has␣
↪reached the daily amount of upload and download traffic which is 243 of 100 MB!␣
↪The user was rejected!!!
```

- MySQL
- LDAP/ActiveDirectory (connecting to MS AD with PAP)
- User-Auth with SQUID
- One-Time-Password

29.9.3 Installation and Configuration

- Navigate to **System > Packages**, **Available Packages** tab.
- Click  at the end of the row for **freeradius3**.
- Confirm the installation.
- Monitor the progress as it installs.

After Installation, the service may be configured at **Services > FreeRADIUS**.

- Select the **interface(s)** on which the RADIUS server should listen on.
- Configure the **NAS/client(s)** from which the RADIUS server should accept packets.
- Add the **user(s)** who should have access.

After this, have a look at the pfSense® syslog. There should be the following:

```
radiusd[16634]: Ready to process requests.
radiusd[16627]: Loaded virtual server
```

Troubleshooting RADIUS Authentication

When attempting to authenticate against a RADIUS server, errors may be encountered in the logs that prevent it from working properly. Here are some errors and how to resolve them:

```
mpd: [pt0] RADIUS: RadiusSendRequest: rad_init_send_request failed: -1
```

- This appears to happen when the RADIUS shared secret contains special characters. Try again with an alphanumeric shared secret.

Get FreeRADIUS Status Server Updates

The status server will give lots of information about the FreeRADIUS server. Many stats are shown about Accounting-Packets, dropped packets and much more. To enable status server and request information from the server do the following:

- Setup an interface with **Interface-Type**: *status* and a free port. The default port for RADIUS accounting is `1813`.
- Setup a NAS/Client with **IP-Address**: `127.0.0.1` and a password. Password *testing123* will be used in this example.
- SSH to the pfSense firewall and enter the following command on the command line:

```
echo "Message-Authenticator = 0x00, FreeRADIUS-Statistics-Type = All" | \
↪radclient localhost:1813 status testing123
```

The output should look like this:

```
Received response ID 223, code 3, length = 140
       FreeRADIUS-Total-Access-Requests = 1
       FreeRADIUS-Total-Access-Accepts = 0
       FreeRADIUS-Total-Access-Rejects = 14
       FreeRADIUS-Total-Access-Challenges = 0
```

(continues on next page)

(continued from previous page)

```
        FreeRADIUS-Total-Auth-Responses = 14
        FreeRADIUS-Total-Auth-Duplicate-Requests = 0
        FreeRADIUS-Total-Auth-Malformed-Requests = 0
        FreeRADIUS-Total-Auth-Invalid-Requests = 0
        FreeRADIUS-Total-Auth-Dropped-Requests = 0
        FreeRADIUS-Total-Auth-Unknown-Types = 0
```

To request other status updates, replace **FreeRADIUS-Statistics-Type = 1** from the command above with another value. More values can be found in this path on the pfSense firewall:

```
/usr/local/share/freeradius/dictionary.freeradius
```

29.10 FRR Package

The FRR package manages dynamic routing for the firewall. Dynamic routing refers to routes that are capable of changing, generally due to routing protocols exchanging routing information with neighboring routers.

Unlike static routes, dynamic routing does not require remote network destinations and gateways to be hardcoded in the configuration. Routes and gateways are automatically determined by the protocol instead.

Currently the FRR package supports the following dynamic routing protocols:

Border Gateway Protocol (BGP) BGP routes between autonomous systems, connecting to defined neighbors to exchange routing and path information. BGP supports IPv4 and IPv6.

Open Shortest Path First v2 (OSPF) OSPF is a link-state routing protocol capable of automatically locating neighboring IPv4 routers within an autonomous system, typically with multicast, and exchanges routing information for networks reachable through each neighbor. OSPF v2 only supports IPv4.

Open Shortest Path First v3 (OSPF6) Similar to OSPF v2, but for IPv6 networks.

29.10.1 FRR Global Settings

The **Global Settings** area in FRR defines options which control the behavior of FRR in general, options which can be utilizied by multiple routing protocols, and options which are not specific to a routing protocol.

These options include **Access Lists**, **Prefix Lists**, and **Route Maps**. These mechanisms allow for fine-tuning dynamic routing behavior.

The options here primarily govern the bahvior of the `zebra` and `staticd` daemons in FRR.

FRR Global Settings Configuration

Configuration of FRR Global Settings is performed at **Services > FRR Global/Zebra**, on the **Global Settings** tab in FRR. From other areas of FRR, these settings can also be reached by the **[Global Settings]** tab.

General Options

Enable Master enable option for FRR. When unchecked, all of FRR is disabled, including individual routing daemons.

Default Router ID The **Router ID** is an IPv4-address formatted string which uniquely identifies this router. Typically it is set to the LAN address of a router or another unique interface address.

This default router ID is used by FRR when a per-protocol router ID is not set.

Master Password The password used by FRR for accessing the management daemons internally. This is not typically used by humans interacting with the daemons in the shell (e.g. via `vtysh`) but only internally in FRR.

Encrypt Password When set, encrypts passwords in output from FRR.

Ignore IPsec Restart When checked, IPsec restarts cause no action to be taken by FRR. When unchecked, IPsec VTI interfaces will be reset in FRR when IPsec restarts. This reset can prevent routes from becoming inactive in the routing table after IPsec VTI interface events.

CARP Status IP Used to determine the CARP status when using FRR with certain high availability setups. When the selected CARP vhid is in BACKUP status, FRR will not be started. This check is also made when a CARP VIP transitions to a new status, and the FRR daemons will be stopped or started appropriately to match the VIP status.

Logging

The dynamic routing manager daemon can send log messages to a file, via syslog, or both.

Syslog Logging Instructs FRR to send its log messages to syslog.

Package Logging Level Controls the verboseness of FRR package scripts

Normal Typical log messages.

Extended Detailed log messages, which may include debugging information.

Modules

The **Enable SNMP AgentX** option in this section controls whether or not data from FRR will be available through the NET-SNMP package.

Note: This feature is not compatible with the `bsnmp` daemon included with the firewall, only the NET-SNMP package.

Route Handling

The options in this section influence FRR global routing behavior. For example, it can setup special automatic lists to control route acceptance and also to setup FRR-based static routes (e.g. `staticd`). These are different than static routes managed in the firewall GUI directly (*Static Routes*).

Do Not Accept When set, routes matching the **Subnet** exactly will not be accepted from routing protocols.

Null Route When set, traffic from hosts inside the defined **Subnet** will never be routed. The traffic will be dropped when it arrives at the firewall.

This option takes precedence over other routing options.

Subnet An IPv4 subnet or IPv6 prefix for this entry.

Static Route Target A list of available system gateways, BGP neighbors, and interfaces which can be used as a destination for this route entry. Selecting an entry from the drop-down turns this entry into an FRR static route.

Force Service Restart

By default, FRR attempts to stay running and enact changes in a dynamic way so that there is no loss of service when possible.

Certain changes may necessitate a full restart of FRR, which can be done with the **Force Service Restart** button in this section.

Access Lists

Access list entries determine if networks are allowed or denied in specific contexts used in various routing daemons. For example, an access list may be used to determine if a route is accepted or rejected, or for limiting routes distributed to neighbors.

Access lists are managed on the **Access Lists** tab under **Services > FRR Global/Zebra**.

Access List Configuration

To create a new access list, click  **Add** from the **Access Lists** tab.

The top section of the page sets data about the access list itself:

Type The type of access list, can be one of:

Standard A standard access list can match source addresses only.

Extended An extended access list can match source or destination addresses.

Zebra A Zebra access list is similar to an **Extended** list, but supports IPv6.

IP Version The IP version to match using this access list, either `IPv4` or `IPv6`.

Name The name of this access list, which will be visible in drop-down lists throughout FRR where access lists can be selected.

The allowed names depend upon the chosen type, and are limited to:

- `1-99` or `1300-1999` for standard access lists.
- `100-199` or `2000-2699` for extended access lists.
- Text names for zebra access lists.

Description A text comment to describe this access list.

Access List Entries

The **Access list entries** list contains rules which govern the behavior of the list. An access list can have multiple rules.

To add more entries to the list, click **Add**.

Sequence The order of entries inside access lists is important, and the order is determined by this sequence number.

Each rule in an access list must have a unique sequence number. Best practice is to leave gaps in the sequence to allow for adding rules in the future. For example, use `10`, `20`, `30`, rather than `1`, `2`, `3`.

> **Warning:** The order of rules displayed in the GUI may be different than the order set by the **sequence** numbers. The **sequence** number order is the true order in which rules are evaluated.

Action The action to take for this rule, either `permit` or `deny`.

Source Network The source IP prefix to match for this rule, given in network/prefix notation. For example, `192.168.0.0/16`.

Source Any When set, the **Source Network** is ignored and any source will match the rule.

Destination Network The destination IP prefix to match for this rule, given in network/prefix notation. For example, `192.168.0.0/16`.

Destination Any When set, the **Destination Network** is ignored and any destination will match the rule.

Exact Will only match if a network prefix matches exactly, rather than matching networks contained within the specified prefix.

Prefix Lists

Prefix List entries determine parts of networks which can be allowed or denied in specific contexts used in routing daemons. For example, a prefix list may be used to match specific routes in a route map.

Prefix lists are managed on the **Prefix Lists** tab under **Services > FRR Global/Zebra**.

Prefix List Configuration

To create a new prefix list, click **Add** from the **Prefix Lists** tab.

The top section of the page sets data about the prefix list itself:

IP Type The IP version to match using this access list, either `IPv4` or `IPv6`.

Name The name of this prefix list, which will be visible in drop-down lists throughout FRR where prefix lists can be selected.

Description A text comment to describe this prefix list.

Prefix List Entries

The **Prefix list entries** list contains rules which govern the behavior of the list. A prefix list can have multiple rules.

To add more entries to the list, click  **Add**.

Sequence The order of entries inside prefix lists is important, and this order is determined by a sequence number.

Each rule in a prefix list must have a unique sequence number. Best practice is to leave gaps in the sequence to allow for adding rules in the future. For example, use `10`, `20`, `30`, rather than `1`, `2`, `3`.

> **Warning:** The order of rules displayed in the GUI may be different than the order set by the **sequence** numbers. The **sequence** number order is the true order in which rules are evaluated.

Action The action to take for this rule, either `permit` or `deny`.

Network The network prefix to match. This may optionally be bound by **Minimum Prefix** (lower bound) or **Maximum Prefix** (upper bound) size limit. When no upper or lower bound is set, the prefix will be matched only exactly as given. Setting bounds allows a prefix list to also match more specific routes which are a contained within the given prefix.

Any When set, matches any prefix.

Minimum Prefix Also known as `ge`. Sets a lower bound for the prefix length. This must be greater than the prefix length given in **Network**, and less than or equal to the value of **Maximum Prefix**, if present.

Maximum Prefix Also known as `le`. Sets an upper bound for the prefix length. This must be greater than the prefix length given in **Network**, and greater than or equal to the value of **Minimum Prefix**, if present.

Prefix List Examples

For example, the following prefix list will match any of the RFC1918 networks:

- Sequence: 10, Action: Permit, Network: `10.0.0.0/8`, Maximum Prefix: `32`
- Sequence: 20, Action: Permit, Network: `172.16.0.0/12`, Maximum Prefix: `32`
- Sequence: 30, Action: Permit, Network: `192.168.0.0/16`, Maximum Prefix: `32`

For each of these entries, the prefix list will match based on the bits specified in the prefix. A match will occur for any network included in the specified range. For example, `10.0.0.0/8` with a **Maximum Prefix** of `32` means a route for any smaller network inside `10.0.0.0/8` will also match, so long as the prefix length is less than `32`. So `10.2.0.0/16` will also match this entry, as will `10.34.157.82/32`. Taken as a whole, this prefix list will match not only the list of RFC1918 networks exactly, but any smaller network wholly contained inside.

As another example, consider this rule instead:

- Sequence: 10, Action: Deny, Network: `10.0.0.0/8`, Minimum Prefix: `24`, Maximum Prefix: `32`

This matches routes for networks inside of `10.0.0.0/8` with a prefix length greater than or equal to `24` but less than or equal to `32`. Meaning it will **not** match larger networks such as `10.2.0.0/16` but it will match more specific

networks such as `10.2.56.128/29` anywhere inside the `10.0.0.0/8` address space. This type of rule can be used to exclude small prefixes from being matched by a route map, for example.

Dynamic Routing Route Maps

Route maps are a powerful mechanism which can match or set various values for use by routing daemons, especially BGP. A route map can match based on criteria such as those set by *Access Lists* and *Prefix Lists*, among others. Route maps can control, for example, whether or not specific routes are accepted from neighbors, or whether or not specific routes are distributed to neighbors. They can also adjust various properties of routes, which largely depends upon the context in which they are used, such as for BGP or OSPF.

Route maps are managed on the **Route Maps** tab under **Services > FRR Global/Zebra**.

Route map entries are complex, and multiple entries can be combined by using the same name on more than one entry, but with different sequence numbers to control the order in which the route map entries are processed by FRR.

Route Map Configuration

To create a new route map, click  **Add** from the **Route Maps** tab.

The **General Options** section of the page sets data about this route map entry:

Name The name of this route map entry.

> **Note:** The same name can be used for multiple entries, but each entry using the same name must use a unique sequence number.

Description A text description of this route map

Action The action taken by this route map, either `permit` or `deny`.

permit When an entry is matched and permitted, the "set" actions of a route map are carried out, if present, and then *Logic Control* entries, if present, are performed. The route will be allowed unless the control flow ultimately prevents that from happening.

deny When an entry is matched and denied, the route is not allowed.

Sequence The sequence number of this route map.

The order of entries inside route maps is important, and this order is determined by a sequence number.

Each entry in a route map must have a unique sequence number. Best practice is to leave gaps in the sequence to allow for adding entries in the future. For example, use `10`, `20`, `30`, rather than `1`, `2`, `3`.

> **Warning:** The order of entries displayed in the GUI may be different than the order set by the **sequence** numbers. The **sequence** number order is the true order in which rules are evaluated.

Route Map Contents

The remaining sections on the page control what this route map entry will do. There are numerous options available, from control and logic flow, to matching, setting, and altering routes.

Generally speaking, when an option in the remainder of the page is set to *None*, it will be ignored or have no effect.

Due to complexity, these options are broken up until multiple sections.

Logic Control

Call Route Map Will immediately process the selected route map. If the called route map returns `deny`, then processing is stopped and the route is denied.

Exit Action

next Proceeds to the next rule in the route-map

<sequence> number Skips to the rule with the given sequence number in this route map.

Access Lists

Sets an *Access List* used to match this route map entry.

Prefix Lists

Sets a *Prefix List* used to match this route map entry.

Next Hop

Controls operations matching or setting the next hop for a route.

Next Hop Action Chooses between actions to take for the next hop of a route.

Match Peer Matches a specific next hop peer (e.g. BGP Neighbor).

Match ACL Matches based on a specific *Access List*.

Match Prefix List Matches based on a specific *Prefix List*.

Set (Peer Only) Changes the next hop on a route to the specified peer (e.g. BGP Neighbor).

Peer Specifies a peer for **Next Hop Action** when it is set to **Match Peer** or **Set (Peer Only)**.

Local (match only) Matches a route when its next hop is this firewall.

Unchanged (set only) Leaves the next hop unchanged.

Peer Address (set only) For inbound IPv4 routes received from a neighbor, sets the next-hop to the address of the neighbor. For outgoing routes this is the local address used to establish an adjacency with the neighbor.

<Neighbor> A list of available peers fills out the list. Selecting an entry uses that specific peer to match or set.

ACL Specifies an *Access List* used to match the next hop value when **Next Hop Action** is set to **Match ACL**.

Prefix List Specifies an *Prefix List* used to match the next hop value when **Next Hop Action** is set to **Match Prefix List**.

Metric

Match or set the metric of a route.

Metric Action Chooses between actions to take for the metric of a route.

Match Matches the given metric value.

Set Sets the MED value for routes. When this router has multiple links to the same AS, the MED value influences which path the router will prefer. The router will prefer to use links with a lower MED value. Adding a + before the metric value will result in a relative adjustment instead of setting an absolute value.

Set OSPF6 External Type 1 Metric Similar to above, but only operates on the OSPF6 External Type 1 Metric.

Set OSPF6 External Type 2 Metric Similar to above, but only operates on the OSPF6 External Type 2 Metric.

Metric Value The metric value to match or apply. When setting a metric, the value may be `+rtt`, `-rtt`, + or − value offset, or a specific metric.

Weight

Sets the weight of the route to the supplied value. When a remote AS is reachable via multiple paths through other intermediate AS neighbors, the router will prefer to use a higher weight path to reach it.

Local Preference

The options in this section will either match or set the BGP local preference value of a route using the given **Local Preference** value.

BGP AS Paths

Matches or sets a *BGP AS Path*.

AS Path Action

Match AS Path Match based on the *BGP AS Paths* selected in **Match AS Path** below.

Set Exclude Excludes the AS numbers specified in **Set AS List** from the path of the route.

Set Prepend Prepends the AS numbers specified in **Set AS List** to the AS path.

Set Prepend Last-AS Prepends the AS number specified in **Set AS List** to the leftmost end of the path.

Match AS Path The specific *BGP AS Path* to match.

Set AS List A list of *BGP AS Path* entries to apply to the route.

BGP Communities

Matches or sets *BGP community* values in routes.

Community Action

Match Match based on community value in **Match Community**.

Match Exact Match, but only if the community value matches exactly, rather than being part of a list.

Set Sets the BGP community value to the list in **Set Community**.

Match Community

internet, no-export, no-advertise, local-as Match one of the well-known communities.

<Community Name> Match a community defined at *BGP Community Lists*.

Set Community When setting a community, this is a **space-separated** list of communities in `AS:VAL` format, or a well-known community: `internet`, `no-export`, `no-advertise`, or `local-as`. Can also be set to `none` to remove BGP community values entirely.

Additive Adds the specified community value to the route without replacing the existing values.

Origin

Origin Action Match or set based on the origin (source) of the route.

Origin Name

Remote EGP Routes from Exterior Gateway Protocols (e.g. BGP).

Local IGP Routes from Interior Gateway Protocols (e.g. OSPF).

Unknown Heritage (Incomplete) Routes from unknown sources.

Source Protocol

Matched based on a specific route source protocol from a list of possible options.

Note: Not all options in the list are supported by the FRR package currently.

Tags

Tag Action

Match Match a tag value set by another route map rule.

Set Set a tag value to be matched by another route map rule.

Tag Value The specific tag value to match or set. This value is an integer from `1-4294967295`.

RPKI

Matches based on the RPKI state.

Prefix Not Found The prefix is not present in the configuration.

Invalid Prefix The prefix is known but failed validation.

Valid Prefix The prefix is known and passed validation.

Route Map Examples

This example creates a route map to control which routes will be sent to peers via BGP. The first rule prevents any route from sending if it matches entries in the `RFC1918` prefix list. The second rule allows routes that match networks listed in the `MY-ROUTES` prefix list. This ensures that even if other mechanisms would try to export routes to peers, that no routes to private networks are leaked.

- Name: EBGP-OUT, Sequence: 10, Action: Deny, Match Prefix List: RFC1918
- Name: EBGP-OUT, Sequence: 20, Action: Permit, Match Prefix List: MY-ROUTES

FRR Status

The status page for FRR can be found at **Status > FRR** or on the **Status** tab.

Accessing the **Status** tab from **Global Settings** shows an overview of FRR and basic information from all active routing protocols.

The protocol and daemon-specific tabs (Zebra, BGP, OSPF, OSPF6, BFD) display more detailed information related to each area.

The **Configuration** tab shows the current configuration file (`frr.conf`).

See also:

- *BGP Status*
- *OSPF Status*

Output Limits and Filtering

Most sections on the status page limit the amount of data shown to ensure that the GUI does not get overwhelmed when dealing with large amounts of data, such as from full BGP feeds.

There are two options which govern the output shown in each area of the status page:

Display XX of YY items This option controls the number of items the GUI will display for that area. There are several values to choose from as well as an **all** selection which displays everything.

> **Warning:** Do not select **all** when using full BGP feeds or other sources of large routing data. Attempting to display that much data will cause the status page to fail, or will drastically reduce performance.

Filter expression This option filters the output so it only includes lines which match the given expression. This can be a simple string to match or a regular expression pattern.

Zebra Status

The `zebra` daemon orchestrates all of the other protocol and related daemons in FRR. As such, it has general information about the operation of FRR in general, plus information which spans across multiple routing protocols.

Zebra Routes The complete IPv4 routing table known to `zebra`, including notations for routes sent from and received through each protocol.

The first column of output contains codes which are explained in the output header. These codes can also explain which routes are associated with each type of :ref:`dynamicrouting-protocol-lists`.

Note: The output here may differ from the OS routing table visible at **Diagnostics > Routes** as not every route in FRR may be set to be active there.

Example entries:

- `B>* 10.7.0.0/16 [20/0] via 10.6.106.2, ipsec4000, weight 1, 4d01h00m`
  - This route was received from BGP (`B`), selected for use (`>`) and is in the firewall routing table (FIB, `*`).
- `O 10.3.0.0/24 [110/10] is directly connected, vmx1, weight 1, 5d15h55m`
  - This route is being advertised to OSPF neighbors (`O`, but directly connected).
- `O>* 10.20.0.0/16 [110/20] via 10.3.111.2, ipsec4000 onlink, weight 1, 4d01h39m`
  - This route was received from OSPF (`O`), selected for use (`>`) and is in the firewall routing table (FIB, `*`).

Zebra IPv6 Routes Same as **Zebra Routes** but for IPv6 routes.

Zebra CPU CPU usage statistics for the `zebra` daemon and its various threads.

Zebra Interfaces Information about system interfaces as seen by `zebra`. This may also include information relevant to interface participation in specific routing protocols.

Zebra Memory Memory usage statistics for various data used by `zebra`.

29.10.2 Border Gateway Protocol

Border Gateway Protocol (BGP) is a dynamic routing protocol used between network hosts. BGP routes between autonomous systems, connecting to defined neighbors to exchange routing information.

BGP can be used for exterior routing (ebgp) or interior routing (ibgp), routing across Internet circuits, private links, or segments of local networks.

BGP Required Information

Before starting, take the time to gather all of the information required to form a BGP adjacency to a neighbor. At a minimum, FRR will need to know these items:

Local AS Number The autonomous system (AS) number for this firewall. This is typically assigned by an upstream source, an RIR, or mutually agreed upon by internal neighbors.

Local Router ID Typically the highest numbered local address on the firewall. This is also frequently set as the internal or LAN side IP address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

Local Network(s) The list of networks that are advertised over BGP as belonging to the Local AS. For external BGP, this is typically the IP address block allocated by the RIR. For internal BGP, this may be a list of local networks or a summarized block.

Neighbor AS Number The autonomous system number of the neighbor.

Neighbor IP Address The IP address of the neighboring router.

The example in this section uses the following values:

Table 1: Example BGP Configuration

Item	Value
Local AS Number	65014
Local Router ID	10.14.0.1
Local Network(s)	10.14.0.0/16
Neighbor AS Number	65002
Neighbor IP Address	203.0.113.2

BGP Example Configuration

The following example configures a BGP adjacency to a neighbor using the settings from *Example BGP Configuration*.

Assumptions

This example makes a few assumptions for brevity and to keep the example simple, including:

- The remote peer is already configured for BGP with equivalent settings.
- Transit to the peer across a directly attached shared network is already configured, for example over a VPN, shared network segment, or peer-to-peer link.
- Firewall rules pass BGP traffic on TCP port 179 between the peers.

Example Configuration

Route Map for Peer Filtering

Before configuring BGP, add a route map to match any routes so it can be used by FRR to allow exchanging all routes with the peer.

Warning: This basic example replicates previous FRR behavior which allowed any routes to be exchanged with a peer. This is convenient, but not secure. For increased security, create a set of route map entries which ensure that only expected routes are sent and received where possible.

- Navigate to **Services > FRR Global/Zebra**, **Route Maps** tab
- Click  **Add**
- Set the following options:

  Name `ALLOW-ALL`

  Description Match any route

  Action Permit

  Sequence 100

FRR BGP Configuration

- Navigate to **Services > FRR BGP**
- Set the following options:

  Enable *Checked*

  Local AS `65014`

  Router ID `10.14.0.1`

  Networks to Distribute `10.14.0.0/16`
- Click **Save**
- Navigate to the **Neighbors** tab
- Click  **Add**
- Set the following options:

  Name/Address `203.0.113.2`

  Remote AS `65002`

  Route Map Filters Set both **Inbound** and **Outbound** to `ALLOW-ALL`

FRR Global Configuration

- Navigate to the **[Global Settings]** tab
- Set the following options:

  Enable *Checked*

  Master Password Create a random string to use
- Click **Save**
- Navigate to the **Status** tab

- Confirm that the BGP neighbor is present and its routes are in the table

FRR BGP Configuration

The FRR BGP service contains numerous methods to configure and fine-tune BGP routing behavior. Due to this complexity, the topic has been split into several sections. Read through each section before attempting to create a new BGP configuration.

BGP Tab Configuration

BGP Router Options

Enable Master enable switch for BGP routing. When checked, FRR will start the BGP routing daemon and attempt to use the BGP settings in this section.

Log Adjacency Changes When set, BGP neighbor adjacency changes will be written via syslog.

Local AS **Required**. The autonomous system (AS) number for this firewall. This is typically assigned by an upstream source, an RIR, or mutually agreed upon by internal neighbors.

Router ID Typically the highest numbered local address on the firewall. This is also frequently set as the internal or LAN side IP address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

Timers

Keep Alive Interval Configures the intervals between keep alive messages.

Hold Time How long to wait for a response before considering the peer unreachable.

Update Delay Keeps BGP in a read-only mode for the specified time after the daemon restarts or peers are cleared.

Peer Wait The amount of time to wait for peers to reach an established state. This starts the same time as the **Update Delay** and allows FRR to end the update delay early if peers are available within the given time period.

Disable Default IPv4 Unicast When unchecked, FRR assumes the peer supports IPv4 unicast in all cases, even when the neighbor is connected over IPv6.

Modules

Enable SNMP AgentX Enable agentx support for accessing FRR BGP data via SNMP with the net-snmp package.

Enable BGP RPKI Enable BGP Resource Public Key Infrastructure.

Global Neighbor Shutdown

Global Neighbor Shutdown When checked, **all** neighbors are placed into an administratively shutdown state.

Message An optional message sent to BGP peers when in this shutdown state.

Graceful Restart/Shutdown

Disable BGP Graceful Restart Globally disable graceful restart functionality in both restart and helper mode.

Preserve FW State If checked, sets the forwarding state (F) bit indication that the FIB is preserved while performing a graceful restart.

Timers

Stale Path Time The time (in seconds) FRR will to retain stale paths from a restarting peer.

Restart Time The time (in seconds) to wait before deleting stale routes unless a BGP open message is received.

Select Defer Time The time (in seconds) FRR defers the route selection process after it restarts.

RIB Stale Time The time (in seconds) stale routes are retained in the RIB.

Enable BGP Graceful Shutdown When set, BGP graceful shutdown is enabled.

RPKI Timers

Configures timers for *BGP RPKI*.

Polling Period The time (in seconds) FRR waits until it queries the cache for updated data.

Expire Interval The time (in seconds) after which FRR will expire RPKI cache data.

Retry Interval The time (in seconds) at which FRR will retry connecting to an RPKI cache server after a connection failure.

Network Distribution

These options control networks for which FRR will distribute or redistribute routes to peers. Peers will be informed to reach these networks through this router.

Redistribute Option Choices Each option in this section may be set to one of the following choices:

No Does not distribute routes from this source.

IPv4 Distributes only IPv4 routes from this source.

IPv6 Distributes only IPv6 routes from this source.

IPv4+IPv6 Distributes both IPv4 and IPv6 routes from this source.

<Route Map Name> Filters distribution of routes by use of the named *route map*.

Redistribute Local These networks are considered local to the router.

Redistribute Connected Networks Redistributes routes for networks which are are attached to and present on interfaces of this firewall.

Redistribute FRR Static Routes Redistributes routes for networks defined as *FRR static routes* in **Global Settings**.

Redistribute Kernel Routing Table Redistributes routes for other networks found in the kernel routing table. This includes static routes defined in pfSense (*Static Routes*), as well as automatic static routes setup for other purposes.

Redistribute OSPF These networks can be reached through OSPF neighbors, not directly on this firewall.

Redistribute OSPF Routes to BGP Neighbors Redistributes routes for networks reachable through IPv4 OSPF neighbors.

Redistribute OSPFv3 Routes to BGP Neighbors Redistributes routes for networks reachable through IPv6 OSPF6 (OSPFv3) neighbors.

Networks To Distribute A manual list of networks that are advertised over BGP as belonging to the Local AS. For external BGP, this is typically the IP address block allocated by the RIR. For internal BGP, this may be a list of local networks or a summarized block.

Subnet to Route An IPv4 subnet or IPv6 prefix to advertise to peers.

Note: If this subnet is not in the routing table (e.g. it is a summary or aggregation) then the **Network Import Check** option in *Advanced BGP Configuration* must be *uncheced*.

Route Map A route map to apply to messages advertising this network.

BGP Neighbor Configuration

BGP Neighbors are managed at **Services > FRR BGP** on the **Neighbors** tab.

The **Neighbors** tab contains a list of current neighbors, if any, and controls to manage the entries (e.g. edit, delete).

The  **Add** button creates a new neighbor.

The remaining sections on this page cover the various options available when creating or editing a neighbor entry.

General Options

Name/Address The name of a peer group or IP address of a neighbor.

Enter a text name to define a Peer Group. Enter an IP Address to define a Peer.

Peer groups allow common options to be defined which may then be applied to multiple neighbors without manually placing the options on each neighbor.

Description A text description about this neighbor.

Peer Group A list of existing peer groups to which this neighbor can be added. Can only be used when defining a neighbor by IP address.

Password Sets a password used to secure communication with this neighbor using TCP MD5.

The operating system of the neighbor and its BGP support may restrict which of the password types can be used.

FRR and setkey Bidirectional Configures the password in FRR and at the operating system level in security policies for both inbound and outbound packets.

This is the best option to use when possible as it fully implements TCP MD5 at the operating system level and in FRR. Use this when both neighbors fully support TCP MD5 for both sending and receiving.

FRR and setkey Outbound Configures the password in FRR and at the operating system level in security policies, but only in the outbound direction.

This option does not validate TCP MD5 on inbound packets, but will add TCP MD5 information to packets sent to the neighbor. Use this if the neighbor is unable to properly send TCP MD5 which can be validated by this firewall.

FRR Only Configures the password only in FRR, not at the operating system level.

setkey Only Outbound Configures the password only at the operating system level in security policies, but only in the outbound direction.

setkey Only Bidirectional Configures the password at the operating system level in security policies for both inbound and outbound packets, but not in FRR.

Shutdown

Neighbor Administrative Shutdown When checked, the neighbor will be put into, and kept in, an administratively shutdown state.

Shutdown Message A text message sent to the neighbor while it is administratively shut down.

Auto-Shutdown

RTT If the round-trip time to the neighbor exceeds this value it will be automatically shut down.

Keep alive Count If the neighbor fails to respond to this number of keep alive messages, it will be automatically shut down.

Basic Options

Remote AS Autonomous System (AS) Number for this neighbor. May be an integer from `1-4294967295`, `external`, or `internal`.

Update Source These options control how FRR will communicate with the neighbor.

IP Type Sets the address family of the IP address to which FRR will bind for communicating to this neighbor, either IPv4 or IPv6.

Local Source Sets the specific IP address to use when communicating with the neighbor. This can be an interface address, an IP alias VIP, or a CARP VIP.

Address Family When set, the neighbor is allowed to advertise routes for both IPv4 and IPv6. Otherwise, the type of routes will be restricted to whichever IP type is set for the **Update Source**.

Default Originate These options control whether or not FRR will advertise itself as the default route for this neighbor.

Originate Default to Neighbor Sets the address family for which a default route will be sent to the neighbor, either IPv4, IPv6, or both.

Route Map A route map used to restrict default origination.

Send Community Sends the community attribute to this peer, limited to the specified types.

Next Hop Self Disables next hop calculation for this neighbor and uses the address of this router instead.

Enabled Uses the address of this router as the next hop in routes announced to this peer if they are learned via eBGP.

Force When set, also sets the next hop to the address of this router on reflected routes.

Inbound Soft Reconfiguration Allows the peer to send requests for soft reconfiguration, to apply changes to routes or new attributes without the need for a session reset.

Timers

Keep Alive Interval Configures the interval between keep alive messages to wait for a response from this neighbor before considering the peer unreachable. This overrides the default values set on the BGP server itself.

Hold Time Configures how long to wait for a response from this neighbor before considering the peer unreachable. This overrides the default values set on the BGP server itself.

Connect Timer The amount of time, in seconds from `1-65535`, in which a connection to this peer must be established or else it is considered unsuccessful.

Peer Filtering

These options control which routes may be sent to, or received from, this neighbor.

Note: The current FRR package does not exchange routes with BGP peers by default without being explicitly allowed to do so by a filter. This is secure behavior but requires manually specifying a filter to allow routes to be exchanged.

To replicate the behavior of older FRR versions, add a *route map* to permit all routes (Name: `allow-all`, Action: *Permit*, Sequence: `100`), then set that route map on BGP neighbors for inbound and outbound peer filtering. For increased security, utilize route maps which filter incoming and outgoing routes so they match more strictly.

Distribute List Filter Defines an *access list* which is used by BGP to filter route updates for this peer, in either the inbound or outbound direction.

Prefix List Filter Defines a prefix list which is used by BGP to filter route updates for this peer, in either the inbound or outbound direction.

AS Path Filter Defines an *AS path list* which is used by BGP to filter route updates by AS path in either the inbound or outbound direction.

Route Map Filters Defines a route map which is used by BGP to filter route updates for this peer, in either the inbound or outbound direction.

Unsuppress Route Map Configures a route map which BGP can use to unsuppress routes that would otherwise be suppressed by other configuration settings.

BFD

Configures *Bidirectional Forwarding Detection* (BFD) options for this peer.

BFD Enable Listen for BFD events registered on the same target as this BGP neighbor.

BFD Check Control Plane Failure Allow FRR to write CBIT independence in outgoing BFD packets. Also allow FRR to read both the CBIT value of BFD and lookup BGP peer status. This option allows BFD to ignore down events during a graceful restart of the remote peer if graceful restarts are enabled in BGP. When enabled, if BFD catches a down event it first checks if the BGP peer has requested that local the BGP daemon keep the remote BGP entries marked as stale. In that case it can safely ignore the event to allow the restart to happen gracefully (RFC 4724).

BFD Peer Selects a *BFD peer* to associate with this neighbor.

Graceful Restart

Graceful restart mode for this neighbor, may be one of:

Default Will use the default value for BGP graceful restart from *Graceful Restart/Shutdown*.

Restart Enables BGP graceful restart functionality for this peer.

Helper Enables BGP graceful restart helper only functionality for this peer.

Disable Disables all BGP graceful restart functionality for this peer.

Advanced Options

Weight Applies the given weight to routes received from this peer.

Passive When set, this router will not issue open requests to the neighbor on its own. The BGP daemon will only respond to remote open requests from this neighbor.

Path Advertise

All Paths Advertise all known paths to this peer, instead of only advertising the base path.

Best Path Advertise only the best known base paths for each AS.

Advertisement Interval Minimal time (in seconds) between sending BGP routing updates to this neighbor.

Allow AS Inbound Allows routes to be received from this peer which are from the same AS of this router, but through a different path.

Enabled Always allow.

Only if Origin Accept the AS of this router in an AS path if the route originated in the AS of this router.

Allow <number>x Allowed number of AS occurrences, from `1-10`.

AS Override Override ASNs in outbound updates to this peer if the AS path is identical to the remote AS.

Attribute Unchanged Propagates route attributes to this peer unchanged. This behavior can be optionally restricted to only specific attributes.

Advertise Capability Advertises the selected capabilities to this neighbor, may be one of:

Dynamic Enables negotiation of the `dynamic` capability with this neighbor or peer group.

Extended Next-Hop Enables negotiation of the `extended-nexthop` capability with this neighbor or peer group. This capability can set IPv6 next-hops for IPv4 routes when peering with IPv6 neighbors on interfaces without IPv4 connectivity. This is automatically enabled when peering with IPv6 link-local addresses.

ORF Advertise outbound route filtering capability to this peer.

Disable Capability Negotiation Disables dynamic capability negotiation with the peer. When set, the router does not advertise capabilities, nor does it accept them. This results in using only locally configured capabilities.

Override Capability Negotiation Ignores capabilities sent by the peer during negotiation and uses locally configured capabilities instead.

TTL Security Hops Sets a specific hop count at which neighbors must be reached, rather than the maximum value set by **eBGP multi-hop**.

This cannot be set if **eBGP multi-hop** is set.

Disable Connected Check Disables a check that normally prevents peering with eBGP neighbors which are not directly connected. This enables using loopback interfaces to establish adjacency with peers.

eBGP Multi-Hop The maximum allowed hops between this router and the neighbor, in the range `1-255`. When enabled without a specific value, the default is `1`.

This value cannot be set if **TTL Security Hops** is set.

Enforce eBGP Multi-Hop When set, enforces that neighbors perform multi-hop.

Local AS

Local AS Number Sets the local AS number sent to this neighbor, which replaces the AS number configured on the BGP server itself. By default, this value is prepended to the AS path for routes received from this neighbor or peer group, and is added to the AS path for routes sent to this neighbor or peer group after the AS number from the BGP sever.

Do not prepend eBGP Suppresses prepending this AS number to the AS path for received routes from eBGP.

Do not prepend iBGP Suppresses prepending this AS number to the AS path for received routes from iBGP.

Maximum Prefix Defines the maximum number of prefixes this router will accept from the peer before tearing down the BGP session.

Note: This action is considered harsh and the best practice is to filter received prefixes by other mechanisms such as a `prefix-list` rather than to abruptly break contact in this way.

Maximum Prefix The maximum number of prefixes to allow from the peer, from `1-4294967295`.

Warn Percentage Warning message threshold, from `1-100` percent.

Warn Only Warn the peer when the limit is exceeded, rather than disconnecting.

Restart Interval Restarts the connection after warning limits are exceeded. The restart is performed at the defined interval, in minutes, from `1-65535`.

Maximum Prefix Out Limits the number of prefixes which will be sent to the neighbor by FRR.

Remove Private AS

Remove Outbound Prevents the BGP daemon from sending routes with private AS numbers to this peer.

Apply to All When present, this action applies to all ASNs.

Replace with Local When present, replaces private AS numbers with the AS number of this router.

Route Client

Route Reflector Client Configures this peer as a route reflector client. This allows routes received from peers in the same AS or using iBGP to be reflected to other peers, avoiding the need for a full mesh configuration between all routing peers.

Route Server Client Configures this peer as a route server client. This enables transparent mode, which retains attributes unmodified, and maintains a local RIB for this peer.

Solo Peer Instructs the router to prevent reflection of routes received from this neighbor back to this neighbor. This option is not useful in peer groups with multiple members.

Advanced BGP Configuration

Advanced Options

Default Local Preference Configure default Local Preference value (0-4294967295, higher=more preferred)

Table Map Uses the specified route map to control how routes received from BGP peers are passed to the dynamic routing manager process, and thus, into routing tables.

Advanced Timers

Coalesce Timer Configures the Subgroup coalesce timer, in milliseconds (`1-4294967295`).

Route Map Delay Time to wait (in seconds) before processing route map changes. A value of `0` disables the timer and stops route updates from happening when route maps change.

Dampening BGP route flap dampening (RFC 2439) prevents unstable routers from adversely affecting routing behavior.

Time Penalty Half Life The time duration during which the stability value will be reduced by half if the route is unreachable.

When to Reuse a Route Stability threshold that must be crossed for a route to be reused.

Start Suppressing Route Stability threshold that, when crossed, a route will be suppressed.

Max Time to Suppress Maximum time to suppress a route considered stable.

Advanced Routing Behavior

Disable Fast External Failover Do not immediately reset session if a link to a directly connected external peer goes down.

Network Import Check Checks if a BGP network route exists in IGP before creating BGP table entries.

> **Note:** This should be disabled if this router wants to advertise manually summarized routes that do not exist in the routing table, such as collecting all local routes into a larger network in which they can be contained (e.g. `10.14.0.0/16`).

Reject AS_SET/AS_CONFED_SET Routes Reject incoming and outgoing routes with AS_SET or AS_CONFED_SET type.

Route Reflecting

Route Reflector Outbound Allows attributes modified by route maps to be reflected.

Cluster ID Route reflector cluster ID.

Disable Client-to-Client Disables reflection of routes from one client to another client.

Aggregate Behavior

Configures route aggregation using a list of prefixes. More specific routes contained within the specified prefixes will be aggregated into the larger prefix, minimizing the set of networks advertised to peers.

Aggregate Address The prefix to aggregate.

Generate AS Set When present, routes for the specified prefix will include an AS set. An AS set is a collection of AS numbers for which routes have been aggregated. This allows peers to detect routing loops, duplicate routes, and so on.

Summary Only When present, aggregated routes for this prefix will not be announced, so peers only see the aggregate prefix and not the component networks.

Multi-Exit Discriminator

Deterministic MED Determine route selection locally, even when MED values are present. Picks the best MED path from neighbor advertisements.

Always Compare MED Instructs the BGP daemon to always consult MED values in routes, no matter which AS the routes were received through.

Max MED

Administratively applied max MED (Indefinite) Sends a MED value of `4294967294` at all times.

Definite admin max MED Sends this value at all times instead of the default `4294967294`.

Time Period for Max MED on startup Sends a MED value of `4294967294` at startup for this number of seconds.

Max MED used during startup Sends this value at startup instead of the default `4294967294`.

Confederation

AS Confederation Configures an AS number for the entire group of iBGP routers participating in confederation.

Confederation Peers Configures the sub-AS number for the subset of peers inside a group of iBGP routers participating in confederation.

Distance

Administrative Distance Manually configures the administrative distance for a given prefix

Define The administrative distance for this prefix, from `1-255`.

IP Source Prefix The IP prefix to which this distance will be applied.

Access List An access list which can be used to apply the distance to only a subset of the configured prefix.

BGP Distance Configures distance values which control how BGP will treat routes based on the length of their AS path.

AS External Routes The distance at which routes are considered external, from `1-255`.

AS Internal Routes The distance at which routes are considered internal, from `1-255`.

Local Routes The distance at which routes are considered local, from `1-255`.

Best Path Selection

Controls how the BGP daemon determines the best path to a destination.

Compare Path with Confederation Considers the length of confederation path sets and sequences.

Ignore AS Path Ignores AS path lengths when computing the route to a destination.

Multipath Relax

Allow Load Sharing Consider paths of equal length when choosing between multiple paths to a destination, rather than looking for an exact match. This allows load sharing across different AS paths, so long as they are of equal length.

Generate an AS_SET Adds AS set information for aggregate routes.

Compare Router ID Uses the router ID of peers (or originator ID, if present) to break ties when computing paths to a destination based on other information. A lower router ID will win in a tie.

MED Confederation

Compare MED among confederation paths Compare confederation path MEDs

Treat missing MED as least preferred path If a route is missing MED information, it will be considered least preferred.

eBGP

eBGP Nexthop Connected Disable checking if nexthop is an eBGP session.

Enforce First AS Enforce the first AS for eBGP routes

Disable eBGP Require Policy Disable the requirement to apply incoming and outgoing filter to eBGP sessions.

Networking Behavior

Subgroup Packet Queue Maximum size of the subgroup packet queue.

Write Quanta Controls the size of peer update transmissions.

BGP AS Paths

AS Path access lists entries determine if networks are allowed or denied in specific BGP configuration contexts. They are primarily used in BGP route maps, but also can be used in other areas of BGP configuration which accept AS Path lists as parameters.

The order of entries inside an AS Path list is important, and this order is determined by a sequence number. As with other access lists, AS Path access lists implicitly deny anything not matched.

AS Path lists are managed at **Services > FRR BGP** on the **AS Paths** tab.

The order of entries inside an AS Path list is important, and this order is determined by a sequence number.

BGP AS Path Configuration

The **AS Paths** tab contains a list of current AS Path lists, if any, and controls to manage the entires (e.g. edit, delete).

The  **Add** button creates a new AS Path list.

When creating or editing an AS Path list, the following options are available:

Name The name of this BGP AS Path list.

Description A text description of this list (e.g. its purpose)

AS Path Entries A list of AS paths to match in this list. Click  **Add** to create additional rules on the same list.

Sequence The sequence number for this rule, which controls the order in which rules are matched inside this AS Path list. Each rule must have a unique sequence number. Best practice is to leave gaps in the sequence to allow for adding rules in the future. For example, use `10`, `20`, `30`, rather than `1`, `2`, `3`.

Action The action taken when this AS Path rule is matched, either *permit* or *deny*.

Regular Expression A regular expression pattern which will match on the AS number.

Regular expression patterns support common pattern special characters for matching, but also a special _ character. The _ character matches common AS delimiters such as start of line, end of line, space, comma, braces, and parenthesis. The _ character can be used on either side of an AS number to match it exactly, such as `_65534_`.

BGP Community Lists

A BGP community, as defined in RFC 1997, is a group of destinations which share common properties. Community Lists define sets of community attributes which the BGP daemon can use to match or set community values in routing updates. BGP communities determine AS membership and priority values in BGP-specific contexts such as route-maps.

BGP community lists are managed at **Services > FRR BGP** on the **Communities** tab.

The order of entries inside a community list is important, and this order is determined by a sequence number.

BGP Well-Known Communities

There are several "well-known" communities available for use in Community Lists. Each of these communities have special meanings:

internet A community value of `0`, indicating the Internet as a destination.

no-export Routes received carrying this attribute value must not be exported to routers outside of the current confederation.

no-advertise Routes received carrying this attribute value must not be advertised to any other BGP peer.

local-as Also known as "No Export Subconfed". Routes received carrying this attribute value must not be advertised to any external BGP peer, even those in the same confederation.

blackhole Routes received carrying this attribute should not be routed (e.g. null routed).

graceful-shutdown Indicates support for RFC 8326 Graceful Shutdown, which allows BGP routers to indicate to peers that specific paths can be gracefully shut down rather than abruptly terminated when performing an intentional shutdown.

no-peer Indicates that routes with this community value should not be readvertised to peers (RFC 3765).

BGP Community List Configuration

The **Communities** tab contains a list of current community lists, if any, and controls to manage the entires (e.g. edit, delete). The  **Add** button creates a new community list.

When creating or editing a community list, the following options are available:

Name The name of this BGP community list.

Description A text description of this list (e.g. its purpose)

Community List Type The type of community list to use for this entry, which controls how values are matched.

Standard Matches based on specific values for community attributes, either AS:Value pairs or names of *well-known communities*.

Expanded Matches based on an ordered list using a regular expression. Due to the use of regular expression evaluation, these lists incur a performance penalty.

Community List Entries A list of rules for this community list. Click  **Add** to create additional rules on the same list.

Sequence The sequence number for this rule, which controls the order in which rules are matched inside this community list. Each rule must have a unique sequence number. Best practice is to leave gaps in the sequence to allow for adding rules in the future. For example, use `10`, `20`, `30`, rather than `1`, `2`, `3`.

Action The action taken when this Community List rule is matched, either *permit* or *deny*.

Community The value of the community to match.

Standard Community Lists This is a space-separated list of communities in AS:Value format, or from the *well-known communities* list.

Expanded Community Lists A string containing a regular expression to match against.

Regular expression patterns support common pattern special characters for matching, but also a special _ character. The _ character matches common AS delimiters such as start of line, end of line, space, comma, braces, and parenthesis.

BGP RPKI Cache Servers

Resource Public Key Infrastructure (RPKI) is a means by which FRR can enact Prefix Origin Validation (POV) to ensure that it is talking to the correct origin for a given AS.

This validation is **not** performed by FRR or other routers directly, but by trusted servers which cache the information.

Note: For more details, see RFC 6810 for the protocol and and RFC 6811 for validation.

RPKI happens over a plain TCP connection but FRR can protect this by performing the validation over SSH.

Route maps can be used to filter routes based on a validated origin.

RPKI Cache Servers are managed at **Services > FRR BGP** on the **RPKI Cache Servers** tab.

RPKI Cache Server Configuration

The **RPKI Cache Servers** tab contains a list of current RPKI Cache Servers, if any, and controls to manage the entires (e.g. edit, delete). The  **Add** button creates a new RPKI Cache Server.

When creating or editing an RPKI Cache Server, the following options are available:

Address **Required**. The IP **Address** or hostname of the RPKI Cache Server, and the **Port** number upon which the service is listening.

Preference **Required**. A preference value FRR can use to decide between multiple RPKI Cache Servers.

SSH Options The best practice is to encrypt communication with the RPKI Cache Server using SSH. The remaining options setup an SSH session, and all are optional.

Username The username to use when connecting to the server via SSH.

Private Key Path Full filesystem path to the private key for this router.

> **Warning:** This must not have a passphrase as there is no way to securely store and use a passphrase. Protect the private key file appropriately, but it must also be accessible to FRR.

Public Key Path Full filesystem path to the public key for this router.

Known hosts Path Full filesystem path to a file containing valid public keys for RPKI Cache Servers in SSH `known_hosts` format.

BGP Status

The status page for BGP can be found at **Status > FRR**, **BGP** tab. It can also be accessed by visiting the **Status** tab while configuring BGP options.

See also:

For general FRR status information, see *FRR Status*.

BGP Routes A list of routes being advertised to BGP and received from BGP. The list includes information about the status of the route, its origin, AS path, and metrics.

BGP IPv6 Routes Same as **BGP Routes** but for IPv6 routes.

BGP Summary A high-level overview of BGP operations, including a brief list of neighbors and a summary of their activity and statistics.

BGP Neighbors Detailed information about BGP peers and their status, including the state of the neighbor (e.g. `Established`), its capabilities, and much more.

BGP Peer Groups Information about BGP Peer Groups, if any are in use.

BGP Next Hops A list of known route targets kept in a cache by FRR.

BGP Memory A summary of memory (RAM) consumed by BGP.

29.10.3 Open Shortest Path First v2 (OSPF)

Open Shortest Path First v2 (OSPF) is a link-state routing protocol defined by RFC 2328. OSPF automatically locates neighboring IPv4 routers within an autonomous system, typically with multicast, and exchanges IPv4 routing information for networks reachable through each neighbor.

OSPF is an interior routing protocol (IGP), and facilitates routing between private links or segments of local networks.

OSPF Terminology

OSPF has common terms used throughout this section which can be confusing for those unfamiliar with the protocol.

Area A collection of routers inside an AS, each sharing the same area ID. An Area ID is typically formatted like an IP address in dotted quad notation, `nnn.nnn.nnn.nnn`, but can also be expressed as an unsigned 32-bit integer.

Area Border Router (ABR) A router connected to multiple areas.

Autonomous System Boundary Router (ASBR) A router connected to external networks (outside the area).

Backbone The central area of an AS, typically area `0.0.0.0`. All areas in the AS connect to the backbone through ABRs.

Cost A numeric value assigned to a link between networks, used by OSPF to calculate optimal paths to a destination. Typically higher bandwidth or higher quality circuits will be assigned a low cost, while circuits that are undesirable will be given a high cost. OSPF will prefer to use a route when it has the lowest total cost from a source to a destination.

Designated Router (DR) In a network with multiple routers, one of them will be elected as a designated router (DR) using `Hello` messages. The DR takes on the task of generating LSA messages for the network, among other special duties.

Flooding The mechanism by which OSPF routers distribute link state database information to neighbors.

Hello Special OSPF messages which introduce neighbors to each other. Using these messages, neighbors can discover each other and begin to form routing relationships.

Interior Gateway Protocol (IGP) A routing protocol, such as OSPF, which exchanges information about how to reach networks *inside* an autonomous system.

Link State Advertisement (LSA) Messages sent by OSPF routers which describe the state of network links, or the router itself, including information about its interfaces and other neighbors.

Link State Database (LSDB) A database containing the collected LSA messages of all routers and networks in the domain.

Link State Advertisement Message Types

LSA messages each have a type, indicating the information carried within. These types may be referenced throughout this section when describing routing behaviors.

Type 1 - Router LSA Sent by every router in an area. Contains a description of all links on the router, including their state and costs.

Type 2 - Network LSA Sent by the DR for a network. Contains a description of every router attached to the network, including the DR.

Type 3 - Network Summary-LSA Sent by ABRs. Contains a description of destinations outside the current area (inter-area) when the destination is an IP network.

Type 4 - ASBR Summary-LSA Similar to Type 3, but when sent when the destination is an ASBR.

Type 5 - AS-external LSA Sent by ASBRs. Contains a description of destinations outside of this AS. Typically each message only contains information about a single destination.

Type 6 - Multicast Group Membership LSA Not used.

Type 7 - NSSA External Link-State Advertisements Similar to Type 5, but are only exchanged inside an NSSA.

Type 8 - External attribute LSA Carry information from external routing protocols, such as BGP, when such destinations are announced with Type 5 LSAs.

Type 9 - Link Scope Opaque LSA Carries information intended for uses other than OSPF, such as available bandwidth. It is carried through to other routers without being processed by OSPF itself. Type 9 messages are for other routers on the same link.

Type 10 - Area Scope Opaque LSA Similar to Type 9, but flooded to all routers in an area.

Type 11 - AS Scope Opaque LSA Similar to Type 9, but flooded to all routers throughout the AS, except for special areas such as stubs.

Area Types

OSPF Areas can be one of several types which alter their behavior in important ways.

Normal A typical area in which all routers know all possible routes.

Stub Area An area with no external connections. Since traffic passing out of a stub area must pass through an ABR, it only needs to know about routes to the ABR, not beyond the ABR. Routers in a stub area do not receive Type 5 LSAs.

Totally Stub Area Similar to a stub area, but routers also do not receive summary LSA messages except for default route information. As such, they do not receive LSA messages of type 3, 4, or 5.

Not-so-Stubby-Area (NSSA) Similar to a Stub area but it may contain static routes to non-OSPF networks. Routers in an NSSA exchange external routing information in Type 7 LSAs instead of Type 5.

NSSA Totally Stub Area Similar to both NSSA and a Totally Stub area. As such, they do not receive LSA messages of type 3, 4, or 5.

Metric Types

Type 1 or E1 A Type 1 external metric, also known as E1, uses a similar cost calculation to typical link states, where internal and external costs are added together to find the total cost.

Type 2 or E2 A Type 2 external metric, also known as E2, only considers external costs and ignores internal costs.

OSPF Required Information

Before starting, take the time to gather all of the information required to form an OSPF adjacency to a neighbor. At a minimum, FRR will need to know these items:

Local Router ID Typically the highest numbered local address on the firewall. This is also frequently set as the internal or LAN side IP address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

Local OSPF Area A designation for the set of networks to which this router belongs. Can be any number capable of being expressed in dotted quad notation (IPv4 address) or as a 32-bit unsigned integer.

In typical OSPF configurations such as this example, `0.0.0.0` is the backbone area between all routers and each local network has its own area. For simpler deployments, the only area may be `0.0.0.0` on every router and interface. Using a single area disables some features such as route summarization.

OSPF Active Interfaces The interfaces on this router upon which the OSPF daemon will advertise itself and look for neighbors. These interfaces are connected to network segments with other routers. They may be connected to local networks or remote point-to-point links. These interfaces must be configured with IP addresses.

OSPF Active Interface Cost Values OSPF calculates the most efficient way to route between networks based on the total cost of a path from source to destination. Less desirable links (e.g. wireless) can be given a higher cost so that paths over faster networks will be used by traffic unless the preferred path is unavailable. For single connections to other networks, this value is not necessary and may be omitted or set to a simple default such as `5` or `10`.

OSPF Passive Interfaces These interfaces contain networks which should be advertised as reachable through this router, but do not contain other routers.

Summary Routes A list of networks to advertise instead of using networks from directly attached interfaces. This allows many similar routes to be summarized as one larger route if they can all be contained within a larger subnet.

This can only be done when connecting to an ABR in a configuration with multiple areas.

Note: This is optional, but without this in place, every network which must be advertised to neighbors must be attached to this firewall and the interfaces must be added as passive interfaces in OSPF. Using a summary route is cleaner in that it advertises less (one large subnet vs many small subnets) and that it requires less ongoing configuration.

The example in this section uses the following values:

Table 2: Example OSPF Configuration

Item	Value
Local Router ID	10.2.0.1
OSPF Backbone Area	0.0.0.0
Local OSPF Area	0.0.0.2
Active Interfaces (Cost)	OPT1 (10)
Passive Interfaces	LAN
Summary Routes	10.2.0.0/16

OSPF Example Configuration

This example configuration implements a multi-area OSPF setup using the required information from *Example OSPF Configuration*.

Assumptions

This example makes a few assumptions for brevity and to keep the example simple, including:

- The remote peer is already configured for OSPF with equivalent settings.
- Transit to the peer across a directly attached shared network is already configured, for example over a VPN, shared network segment, or peer-to-peer link.
- The link to the peer is capable of handling multicast traffic.
- Firewall rules pass OSPF traffic, which is protocol `89`. It is not TCP or UDP. Firewall rules must allow multicast traffic destinations for OSPF, and it cannot be restricted to specific sources and destinations in this example.

Example Configuration

The OSPF configuration must be done in the following order initially. Later changes may be made in any order. This is because a valid OSPF configuration requires **Interface** tab entries which must exist before the main OSPF settings can be saved.

OSPF Area Configuration

- Navigate to **Services > FRR OSPF**, **Areas** tab
- Click  **Add** to create a new area
- Set the following options:

  Area `0.0.0.0`

  Description `Backbone`
- Click **Save**
- Click  **Add** to create a new area
- Set the following options:

  Area `0.0.0.23`

  Description `Local Area`

  Area Type *Stub Area (stub)*

  Summary Range `10.2.0.0/16`
- Click **Save**

OSPF Interface Configuration

- Navigate to **Services > FRR OSPF**, **Interfaces** tab
- Click  **Add** to create a new interface
- Set the following options:

  Interface *LAN*

  Interface is Passive *Checked*

  Area `0.0.0.23`
- Click **Save**
- Click  **Add** to create a new interface
- Set the following options:

  Interface *OPT1*

  Ignore MTU *Checked*

  Metric `10`

  Area `0.0.0.0`
- Click **Save**

OSPF Configuration

- Navigate to **Services > FRR OSPF**, **OSPF** tab
- Set the following options:

  Enable *Checked*

  Router ID `10.2.0.1`

  Default Area `0.0.0.0`
- Click **Save**

FRR Global Configuration

- Navigate to the **[Global Settings]** tab
- Set the following options:

  Enable *Checked*

  Master Password Create a random string to use
- Click **Save**
- Navigate to the **Status** tab
- Confirm that the OSPF neighbor is present and its routes are in the table

OSPF Configuration

OSPF FRR configuration, as shown in the example, can be fairly straightforward. That said, there are a number of ways to fine-tune the behavior and create complex OSPF routing configurations.

Read through each section before attempting to create a new OSPF configuration.

Note: Though the documentations covers OSPF in *tab* order, OSPF interfaces must be configured before the general OSPF options.

OSPF Tab Configuration

The options on this page configure the general behavior of OSPF in FRR, including various default values used when more specific options are not available.

> **Warning:** When creating an initial OSPF configuration, configure the *interfaces to be used with OSPF* first. The interface configuration must exist before these general OSPF settings can be saved.

General Options

Enable Master enable switch for OSPF routing. When checked, FRR will start the OSPF routing daemon and attempt to use the OSPF settings in this section.

Log Adjacency Changes When set, instructs the OSPF daemon to log changes in neighbor adjacencies. This is useful for tracking changes to neighbor relationships, especially during initial configuration.

Router ID Typically the highest numbered local address on the firewall. This is also frequently set as the internal or LAN side IP address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

SPF Hold Time SPF timers determine when the router will make SPF routing decisions. Lowest time allowed between SPF calculations. Specified in milliseconds from `0-600000`, with a default value of `1000`.

The maximum time is calculated as 10x this value.

SPF calculations are adaptive, and if a new event occurs which would otherwise trigger a calculation before the hold timer expires, then the hold is increased by the **SPF Hold Time** value, up to the maximum. This avoids excessive consecutive recalculations.

SPF Delay Controls timers that determine when the router will make SPF routing decisions. Minimum time after an event occurs before allowing SPF calculation. Lower values will react faster to changes, but can be less stable. Specified in milliseconds from `0-600000`, with a default value of `200`.

Modules

Enable SNMP AgentX Enable agentx support for accessing FRR OSPF data via SNMP with the net-snmp package.

Default Area

Default Area Default OSPF area for this instance of OSPF. Used when an area is required but not defined elsewhere. See *OSPF Area Configuration* for details.

Default Area Type Sets the type for the **Default Area**. See *Area Types* and *OSPF Area Configuration* for details.

OSPF Networks

Warning: This section is **deprecated** and will not be covered here. Define *areas* and use areas on *interfaces* instead. Use summary routes instead, or use route-maps and distribute lists to limit distributed networks.

Route Redistribution

This section controls which, if any, IPv4 routes are redistributed to OSPF neighbors from other sources (*Dynamic Routing Protocol Lists*). OSPF can redistribute IPv4 routes from connected networks, kernel routes, BGP, and FRR static routes.

Redistribute Enables redistribution of routes from the given source.

Metric Advertise the routes from this source as having the given metric.

Metric Type The type of metric, either `1` or `2`. See *Metric Types* for details about each type operates.

Route Map Apply the given route map to the redistributed route advertisements.

Distribute List Applies the given *access list* to routes redistributed from a given route source.

Default Route Redistribution

Redistribute Enables origination of a Type 5 AS-External LSA containing default route information into all areas capable of external routing.

Always Redistribute Always advertise a default route, even when a default route is not present in the local routing table.

Default Metric Advertise the default route as having the given metric.

Default Metric Type The type of metric, either `1` or `2`. See *Metric Types* for details about each type operates.

Route Map Apply the given route map to the outbound default route advertisement.

Advanced

RFC 1583 Compatibility Enables compatibility with the older OSPF standard from RFC 1583, which has been obsoleted by the newer RFC 2328. The specific change this option enables relates to external path preference calculation and routing loop prevention. See RFC 2328 section **G.2** for specific details.

Opaque LSA Enables support for Opaque LSAs, as described in RFC 2370.

Reference Bandwidth A base value, in Mbit/s, which is used when OSPF automatically calculates cost values. The default value is `100` which means that an interface with 100Mbit/s of bandwidth or greater will have a cost of `1`, with lower bandwidth values incurring higher cost values.

All routers in the same area should use the same value, otherwise automatic cost calculations would fail to accurately represent total path costs between routers.

Max Metric

Administratively Enable Max Metric Sets the administrative distance of routes through this router to infinity, so that other routers will avoid using this router to reach other networks. Networks on this router are still reachable. See RFC 3137 for more information.

Startup Seconds Conditionally sets the administrative distance of routes through this router to infinity for a period of time after startup.

This allows other routers in the area to avoid using routes through this router until a full convergence is achieved.

Shutdown Seconds Conditionally sets the administrative distance of routes through this router to infinity for a period of time after shutdown.

This allows other routers in the area to avoid using routes through this router until a full convergence is achieved.

Write Multiplier Number of interfaces processed per write operation, from `1-100`. Default value is `20`.

ABR Type Controls the behavior of Area Border Router (ABR) functionality.

ciscolibm The default behavior of OSPF in FRR, discussed in RFC 3509. This behavior allows an ABR without a backbone connection to act as an internal router for all connected areas.

shortcut Discussed in draft-ietf-ospf-shortcut-abr-02, this behavior allows ABRs to consider summary LSAs from all attached areas, rather than being forced to route through a suboptimal path only because it is shorter.

standard The ABR behavior described in the original OSPF standard. When set, a router attached to multiple areas requires a connection to a backbone. If no backbone is available, traffic attempting to cross areas will be dropped.

OSPF Area Configuration

OSPF Areas are managed at **Services > FRR OSPF** on the **Areas** tab.

The **Areas** tab contains a list of current OSPF areas, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new area.

The remaining sections on this page cover the various options available when creating or editing an area entry.

Area Options

Area A designation for the set of networks. Can be any number capable of being expressed in dotted quad notation (IPv4 address) or as a 32-bit unsigned integer.

In typical OSPF configurations `0.0.0.0` is the backbone area between all routers and each local network has its own area. For simpler deployments, the only area may be `0.0.0.0` on every router and interface. Using a single area disables some features such as route summarization.

Description Some text describing this area.

Area Type Defines how this area behaves. For a list of all types and how they operate, see *Area Types*.

Default Route Cost Sets the cost applied to default route summary LSA messages sent to stub areas.

ABR Shortcut For use with **ABR Type** set to *Shortcut* (*Advanced*), this advertises the area as capable of supporting ABR shortcut behavior (draft-ietf-ospf-shortcut-abr-02).

Authentication

This option enables authentication for this area. Communication from peers must contain the expected authentication information to be accepted, and outgoing packets will have authentication information added.

Authentication passwords and keys are configured by the *OSPF interface settings*.

Authentication Type Sets the type of authentication to use in this area.

None Do not use authentication in this area.

Message Digest (MD5 Hash) Enables MD5 HMAC authentication for this area. This is stronger authentication than simple passwords.

Simple Password Enables simple password authentication for this area.

Route Summarization

Configure summarization of routes inside the given prefixes. Instead of Type 1 (Router) and Type 2 (Network) LSAs, it creates Type 3 Summary LSAs instead.

The options here can be repeated for multiple prefixes. Click  **Add** for each additional prefix.

Summary Prefix The prefix to summarize.

Do Not Advertise Disable advertisement for this prefix.

Cost Apply the specified cost to summarized routes for this prefix.

Substitute Prefix Instead of advertising the first prefix, advertise this prefix instead.

ABR Summary Route Filtering

Export List Uses the given ACL to limit Type 3 summary LSA messages for intra-area paths that would otherwise be advertised. This behavior only applies if this router is the ABR for the area in question.

Import List Similar to `export-list`, but for routes announced by other routers into this area.

Filter List (Out|In) Similar to **Export List** and **Import List** but uses prefix lists instead of ACLs, and can work in either direction.

OSPF Interface Configuration

OSPF interfaces are managed at **Services > FRR OSPF** on the **Interface** tab.

The **Interfaces** tab contains a list of current OSPF interfaces, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new interface.

OSPF must use one or more interfaces to announce itself to neighbors and to receive announcements from neighbors. At least one interface must be configured and active in order to locate neighbors and form an adjacency.

The remaining sections on this page cover the various options available when creating or editing an interface entry.

Interface Options

Interface The interface to use with OSPF. Entries should be added for interfaces which connect to other routers (neighbors) as well as interfaces containing networks which should be advertised to OSPF neighbors.

Description Text describing the purpose of this interface in OSPF.

Network Type Manually configures a specific type of network used on a given interface, rather than letting OSPF determine the type automatically. This controls how OSPF behaves and how it crafts messages when using an interface.

Tip: Most environments will use either *Broadcast* mode (e.g. Ethernet) or *Point-to-Point* mode (e.g. VPNs).

Broadcast Broadcast networks, such as typical Ethernet networks, allow multiple routers on a segment and OSPF can use broadcast and multicast to send messages to multiple targets at once. OSPF assumes that all routers on broadcast networks are directly connected and can communicate without passing through other routers.

Non-Broadcast Non-broadcast networks support multiple routers but do not have broadcast or multicast capabilities. Due to this lack of support, neighbors must be manually configured using the **Neighbor** tab (*OSPF Neighbor Configuration*).

When using this mode, OSPF simulates a broadcast network using Non-Broadcast Multi-Access (NMBA) mode, but transmits messages to known neighbors directly.

Point-to-Multipoint Similar to *Non-Broadcast* mode, but connections to manually configured neighbors are treated as a collection of point-to-point links rather than a shared network. Similar to a point-to-point network, OSPF disables DR election.

Point-to-Point A point-to-point network links a single pair of routers. The interface is still capable of broadcast, and OSPF will dynamically discover neighbors. With this type of network, OSPF disables election of a DR.

Interface is Passive Configures the specified interface as passive. This prevents the interface from actively participating in OSPF, while still allowing OSPF to operate on networks connected to that interface. This is commonly used for local interfaces without other routers attached. OSPF will announce networks attached to passive interfaces as stub links.

Ignore MTU When present, OSPF will ignore the MTU advertised by neighbors and can still achieve a full adjacency when peers do not have matching MTU values.

Tip: If two neighbors are stuck in an `ExStart` state, that is typically from an MTU mismatch. If fixing the MTU mismatch is not viable, set this option on both sides.

OSPF Interface Handling

Metric A manual cost value to apply to this interface, rather than allowing automatic cost calculation to take place.

In situations where multiple paths are possible to the same destination, this allows OSPF to prefer one path over another when all else is equal.

Area This defines the interface as a member of the given area. If this is left blank, FRR will take the value set in *Default Area*.

Accept Filter When set, automatically configures lists behind the scenes to prevent this interface subnet from being advertised to, or received from, OSPF.

This can avoid problems seen in Multi-WAN deployments where there are multiple paths to a remote neighbor and a router can end up learning a route to itself across a VPN link, which is problematic.

Authentication

Configures authentication for OSPF neighbors on this interface. All routers connected to this interface must have identical authentication configurations. This can also be enabled in the area settings.

Authentication Type Sets the type of authentication to use in this area.

None Do not use authentication in this area. This is the default behavior, but may be explicitly configured to override the authentication configured for this area.

Message Digest (MD5 Hash) Enables MD5 HMAC authentication for this area. This is stronger authentication than simple passwords.

Simple Password Enables simple password authentication for this area.

Password Password to use with *Simple Password* or key to use with *Message Digest* authentication.

This value must match all neighbors reachable through this interface. Simple passwords may be up to 8 characters, Message Digest passwords (keys) may be up to 16 characters.

Advanced

Router Priority A priority value, from `0-255`, assigned to this router. When determining which router will become the Designated Router (DR), the router with the highest priority is more likely to be elected as the DR.

The default value is `1`. The value `0` is special and will prevent this router from being chosen as DR.

Retransmit Interval The interval, in seconds from `1-65535`, at which this router will retransmit Link State Request and Database Description messages. This is also known as the `RxmtInterval` timer in OSPF. Default value is `5`.

Hello Interval The interval, in seconds from `1-65535`, at which this router will send hello messages. This is also known as the `HelloInterval` timer in OSPF. Default value is `10`. This timer should be set to the same value for all routers.

A lower value will result in faster convergence times, but will consume more resources.

Dead Interval Time, in seconds from `1-65535`, without communication from a neighbor on this interface before considering it dead. This is also known as the `RouterDeadInterval` timer in OSPF. Default value is `40`. This timer should be set to the same value for all routers.

Minimal Hello When active, the **Dead Interval** is forced to a value of `1` and OSPF will instead send this number of Hello messages each second. This allows for faster convergence, but will consume more resources.

Note: When set, this overrides the values of both **Dead Interval** and **Hello Interval**. Custom values configured in those fields will be ignored by OSPF.

BFD

When set, enables *Bidirectional Forwarding Detection* for OSPF on this interface.

OSPF Neighbor Configuration

OSPF neighbors are managed at **Services > FRR OSPF** on the **Neighbors** tab.

The **Neighbors** tab contains a list of current OSPF neighbors, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new neighbor.

In most cases OSPF neighbors do not need to be added manually. In certain environments with links which do not support multicast, manual neighbor definitions allow FRR to locate neighbors statically. This can be useful to work around limitations imposed by certain point-to-point links or VPN configurations.

The remaining sections on this page cover the various options available when creating or editing a neighbor entry.

Neighbor Options

OSPF Neighbor IPv4 Address The IPv4 address of this OSPF neighbor.

Description Text describing this neighbor.

Neighbor Priority A priority value applied to neighbors in a down state.

Dead Neighbor Polling Interval Time, in seconds, between sending OSPF Hello messages to neighbors in a down state.

OSPF Status

The status page for OSPF can be found at **Status > FRR**, **OSPF** tab. It can also be accessed by visiting the **Status** tab while configuring OSPF options.

See also:

For general FRR status information, see *FRR Status*.

OSPF General General information about the OSPF process, active configuration, and operation.

OSPF Neighbors A list of OSPF neighbors detected by FRR and their status.

> **Tip:** If a neighbor is stuck in a state such as "ExStart", check that the interface MTUs match. Consider checking the interface option to **Ignore MTU** which will bypass this check.

OSPF Routes A list of routes being advertised to OSPF and received from OSPF. The list is broken into three parts:

OSPF Network Routing Table A list of internal networks and their status, how they are reached through neighbors and areas, etc.

OSPF Router Routing Table A list of other routers through which OSPF networks are reached.

OSPF External Routing Table A list of external routes from OSPF.

OSPF Database The OSPF link state database with information about each known router and area.

OSPF Router Database Detailed information about OSPF routers and how they are connected.

OSPF Interfaces Detailed information about interfaces on the firewall and how they are used by OSPF.

OSPF CPU Usage The amount of CPU time used by OSPF for various tasks and threads.

OSPF Memory The amount of memory (RAM) used by OSPF for various tasks and threads.

29.10.4 Open Shortest Path First v3 (OSPF6)

Open Shortest Path First v3 (OSPF6) is defined by RFC 5340 and is similar to OSPF v2, but operates with IPv6 networks. Thus, it is a link-state routing protocol that automatically locates neighboring IPv6 routers within an autonomous system, typically with multicast, and exchanges IPv6 routing information for networks each neighbor.

OSPF6 is an interior routing protocol (IGP), and facilitates routing between private links or segments of local networks.

Terms used in this section are shared with OSPF, and are covered in *OSPF Terminology*.

OSPF6 Required Information

Before starting, take the time to gather all of the information required to form an OSPF6 adjacency to a neighbor. This list is similar to *that of OSPF*. At a minimum, FRR will need to know these items:

Local Router ID Typically the highest numbered local address on the firewall. This is also frequently set as the internal or LAN side IP address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

OSPF6 Area A designation for the set of networks to which this router belongs. Can be any number capable of being expressed in dotted quad notation (IPv4 address) or as a 32-bit unsigned integer.

OSPF6 Active Interfaces The interfaces on this router upon which the OSPF6 daemon will advertise itself and monitor for neighbors. These interfaces are connected to network segments with other routers. They may be connected to local networks or remote point-to-point links. These interfaces only require an IPv6 link local address.

OSPF6 Active Interface Cost Values OSPF6 calculates the most efficient way to route between networks based on the total cost of a path from source to destination. Less desirable links (e.g. wireless) can be given a higher cost so that paths over faster networks will be used by traffic unless the preferred path is unavailable. For single connections to other networks, this value is not necessary and may be omitted or set to a simple default such as `5` or `10`.

OSPF6 Passive Interfaces These interfaces contain networks which FRR will advertise as reachable through this router, but do not contain other routers.

The example in this section uses the following values:

Table 3: Example OSPF Configuration

Item	Value
Local Router ID	10.2.0.1
OSPF6 Area	0.0.0.0
Active Interfaces (Cost)	OPT1 (10)
Passive Interfaces	LAN

OSPF6 Example Configuration

This example configuration implements an OSPF6 setup using the required information from *Example OSPF Configuration*.

Assumptions

This example makes a few assumptions for brevity and to keep the example simple, including:

- The remote peer is already configured for OSPF6 with equivalent settings.
- IPv6 transit to the peer across a directly attached shared network is already configured, for example over a VPN, shared network segment, or peer-to-peer link.
- The link to the peer is capable of handling multicast traffic.
- Firewall rules pass OSPF traffic, which is protocol `89`. It is not TCP or UDP. Firewall rules must allow multicast traffic destinations for OSPF, and it cannot be restricted to specific sources and destinations in this example.

Example Configuration

The OSPF6 configuration must be done in the following order initially. Later changes may be made in any order. This is because a valid OSPF6 configuration requires **Interface** tab entries which must exist before the main OSPF6 settings can be saved.

OSPF6 Interface Configuration

- Navigate to **Services > FRR OSPF6**, **Interfaces** tab
- Click  **Add** to create a new interface
- Set the following options:

  Interface *LAN*

  Interface is Passive *Checked*

  Area `0.0.0.0`
- Click **Save**
- Click  **Add** to create a new interface
- Set the following options:

  Interface *OPT1*

  Ignore MTU *Checked*

  Metric `10`

  Area `0.0.0.0`
- Click **Save**

OSPF6 Configuration

- Navigate to **Services > FRR OSPF6**, **OSPF6** tab
- Set the following options:

  Enable *Checked*

  Router ID `10.2.0.1`

  Default Area `0.0.0.0`
- Click **Save**

FRR Global Configuration

- Navigate to the **[Global Settings]** tab
- Set the following options:

  Enable *Checked*

  Master Password Create a random string to use
- Click **Save**
- Navigate to the **Status** tab
- Confirm that the OSPF6 neighbor is present and its routes are in the table

OSPF6 Configuration

There are a number of ways to fine-tune the behavior and create complex OSPF6 routing configurations. The available configuration parameters are covered throughout this section.

OSPF6 Tab Configuration

The options on this page configure the general behavior of OSPF6 in FRR, including various default values used when more specific options are not available.

> **Warning:** When creating an initial OSPF6 configuration, configure the *interfaces to be used with OSPF6* first. The interface configuration must exist before these general OSPF6 settings can be saved.

General Options

Enable Master enable switch for OSPF6 routing. When checked, FRR will start the OSPF6 routing daemon and attempt to use the OSPF6 settings in this section.

Log Adjacency Changes When set, instructs the OSPF6 daemon to log changes in neighbor adjacencies. This is useful for tracking changes to neighbor relationships, especially during initial configuration.

Router ID Typically the highest numbered local IPv4 address on the firewall. This is also frequently set as the internal or LAN side IPv4 address of a router. It does not matter what this ID is, so long as it is given in IPv4 address notation and does not conflict with any neighbors.

Note: Even though OSPF6 handles IPv6 routing, router IDs are specified using IPv4 addresses in dotted quad notation.

SPF Hold Time SPF timers determine when the router will make SPF routing decisions. Lowest time allowed between SPF calculations. Specified in milliseconds from `0-600000`, with a default value of `1000`.

The maximum time is calculated as 10x this value.

SPF calculations are adaptive, and if a new event occurs which would otherwise trigger a calculation before the hold timer expires, then the hold is increased by the **SPF Hold Time** value, up to the maximum. This avoids excessive consecutive recalculations.

SPF Delay Controls timers that determine when the router will make SPF routing decisions. Minimum time after an event occurs before allowing SPF calculation. Lower values will react faster to changes, but can be less stable. Specified in milliseconds from `0-600000`, with a default value of `200`.

Modules

Enable SNMP AgentX Enable agentx support for accessing FRR OSPF6 data via SNMP with the net-snmp package.

Default Area

Default Area Default OSPF area for this instance of OSPF6. Used when an area is required but not defined elsewhere. See *OSPF6 Area Configuration* for details.

Default Area Type Sets the type for the **Default Area**. See *Area Types* and *OSPF6 Area Configuration* for details.

Route Distribution

The **Distribute Ranges** entries instruct OSPF to associate specific IPv6 prefixes with a given OSPF6 area and to advertise them to neighbors. Additional entries can be created by the  **Add** button.

Note: In most cases these entries are not necessary to add manually, as interface entries will add appropriate statements to distribute interface subnets as needed.

Subnet to Route The IPv6 prefix to advertise.

Area ID The area with which to associate the prefix.

Cost The cost associated with the prefix.

Route Redistribution

This section controls which, if any, IPv6 routes are redistributed to OSPF6 neighbors from other sources (*Dynamic Routing Protocol Lists*). OSPF6 can redistribute IPv6 routes from connected networks, kernel routes, BGP, and FRR static routes.

Redistribute Enables redistribution of routes from the given source.

Route Map Apply the given route map to the redistributed route advertisements.

Route Filtering

Export List Uses the given ACL to limit Type 3 summary LSA messages for intra-area paths that would otherwise be advertised.

Import List Similar to `export-list`, but for routes announced by other routers into this area.

Filter List (Out|In) Similar to **Export List** and **Import List** but uses prefix lists instead of ACLs, and can work in either direction.

Advanced

Reference Bandwidth A base value, in Mbit/s, which is used when OSPF6 automatically calculates cost values. The default value is `100` which means that an interface with 100Mbit/s of bandwidth or greater will have a cost of `1`, with lower bandwidth values incurring higher cost values.

All routers in the same area should use the same value, otherwise automatic cost calculations would fail to accurately represent total path costs between routers.

Distance Sets an administrative distance for routes obtained via OSPF6. This can be configured globally as well as for specific types of OSPF6 routes.

External Distance Sets the administrative distance for external OSPF6 routes.

Inter-area Distance Sets the administrative distance for OSPF6 routes between areas.

Intra-area Distance Sets the administrative distance for OSPF6 routes inside an area.

OSPF6 Interface Configuration

OSPF6 interfaces are managed at **Services > FRR OSPF6** on the **Interface** tab.

The **Interfaces** tab contains a list of current OSPF6 interfaces, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new interface.

OSPF6 must use one or more interfaces to announce itself to neighbors and to receive announcements from neighbors. At least one interface must be configured and active in order to locate neighbors and form an adjacency.

The remaining sections on this page cover the various options available when creating or editing an interface entry.

Interface Options

Interface The interface to use with OSPF6. Entries should be added for interfaces which connect to other routers (neighbors) as well as interfaces containing networks which should be advertised to OSPF6 neighbors.

Description Text describing the purpose of this interface in OSPF6.

Network Type Manually configures a specific type of network used on a given interface, rather than letting OSPF6 determine the type automatically. This controls how OSPF6 behaves and how it crafts messages when using an interface.

Broadcast Broadcast networks, such as typical Ethernet networks, allow multiple routers on a segment and OSPF6 can use broadcast and multicast to send messages to multiple targets at once. OSPF6 assumes that all routers on broadcast networks are directly connected and can communicate without passing through other routers.

Point-to-Point A point-to-point network links a single pair of routers. The interface is still capable of broadcast, and OSPF6 will dynamically discover neighbors. With this type of network, OSPF6 disables election of a DR.

Interface is Passive Configures the specified interface as passive. This prevents the interface from actively participating in OSPF6, while still allowing OSPF6 to operate on networks connected to that interface. This is commonly used for local interfaces without other routers attached. OSPF6 will announce networks attached to passive interfaces as stub links.

Ignore MTU When present, OSPF6 will ignore the MTU advertised by neighbors and can still achieve a full adjacency when peers do not have matching MTU values.

Tip: If two neighbors are stuck in an `ExStart` state, that is typically from an MTU mismatch. If fixing the MTU mismatch is not viable, set this option on both sides.

OSPF6 Interface Handling

Area This defines the interface as a member of the given area. If this is left blank, FRR will take the value set in *Default Area*.

Instance ID An alternate OSPF6 instance identifier for this interface. Typically omitted or set to `0`.

Metric A manual cost value to apply to this interface, rather than allowing automatic cost calculation to take place.

In situations where multiple paths are possible to the same destination, this allows OSPF6 to prefer one path over another when all else is equal.

Advanced

Router Priority A priority value, from `0-255`, assigned to this router. When determining which router will become the Designated Router (DR), the router with the highest priority is more likely to be elected as the DR.

The default value is `1`. The value `0` is special and will prevent this router from being chosen as DR.

Hello Interval The interval, in seconds from `1-65535`, at which this router will send hello messages. This is also known as the `HelloInterval` timer in OSPF6. Default value is `10`. This timer should be set to the same value for all routers.

A lower value will result in faster convergence times, but will consume more resources.

Dead Interval Time, in seconds from `1-65535`, without communication from a neighbor on this interface before considering it dead. This is also known as the `RouterDeadInterval` timer in OSPF6. Default value is `40`. This timer should be set to the same value for all routers.

Retransmit Interval The interval, in seconds from `1-65535`, at which this router will retransmit Link State Request and Database Description messages. This is also known as the `RxmtInterval` timer in OSPF6. Default value is `5`.

BFD

When set, enables *Bidirectional Forwarding Detection* for OSPF6 on this interface.

OSPF6 Area Configuration

OSPF6 Areas are managed at **Services > FRR OSPF6** on the **Areas** tab.

The **Areas** tab contains a list of current OSPF6 areas, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new area.

The remaining sections on this page cover the various options available when creating or editing an area entry.

Area Options

Area A designation for the set of networks to which this router belongs. Can be any number capable of being expressed in dotted quad notation (IPv4 address) or as a 32-bit unsigned integer.

Description Some text describing this area.

Area Type Defines how this area behaves. For a list of all types and how they operate, see *Area Types*.

Note: The list of area types supported by OSPF6 is shorter than OSPF. Consult the GUI drop-down for a current list of supported area types.

Range Overrides

The **Ranges** list overrides the more specific ranges automatically determined from interfaces or by manual entry on the main configuration page. For example, this can be used to summarize routes or to prevent certain prefixes from being advertised.

When summarizing routes, instead of Type 1 (Router) and Type 2 (Network) LSAs, OSPF6 creates Type 3 Summary LSAs instead.

The options here can be repeated for multiple prefixes. Click  **Add** for each additional prefix.

Prefix The prefix to override.

Do Not Advertise Disable advertisement for this prefix.

Cost Apply the specified cost to summarized routes for this prefix.

> **Warning:** An entry can either set **Do Not Advertise** or it can set **Cost**, but it **cannot** set both at the same time.

ABR Summary Route Filtering

Export List Uses the given ACL to limit Type 3 summary LSA messages for intra-area paths that would otherwise be advertised. This behavior only applies if this router is the ABR for the area in question.

Import List Similar to `export-list`, but for routes announced by other routers into this area.

Filter List (Out|In) Similar to **Export List** and **Import List** but uses prefix lists instead of ACLs, and can work in either direction.

OSPF6 Status

The status for OSPF6 contains the same areas as the status for OSPF, but for IPv6 instead. See *OSPF Status* for more information.

See also:

For general FRR status information, see *FRR Status*.

29.10.5 Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is used to detect faults between two routers across a link, even if the physical link does not support failure detection. Even in cases where physical link issues occur and are detected, BFD can coordinate reaction to these failures rather than each component relying on its own failure detection methods.

FRR uses UDP as a transport for BFD between directly connected routers (single hop/next hop) as described in RFC 5880 and RFC 5881.

Each BFD session monitors one link. Multiple BFD sessions are necessary to detect faults on multiple links. BFD sessions must be manually configured between endpoints as there is no method for automated discovery.

When using BFD, both endpoints transmit "Hello" packets back and forth between each other. If these packets are not received within the expected time frame the link is considered down. Links may also be administratively configured as down and will not recover until manually changed.

FRR currently supports BFD integration with BGP, OSPF, and OSPF6.

BFD is configured at **Services > FRR BFD**.

The **BFD** tab contains only the master **Enable** switch for BFD.

Note: The BFD implementation in FRR does not currently support authentication.

BFD Peers

A BFD **Peer** entry defines a relationship between this router and a peer so they can exchange BFD information and detect link faults.

Tip: For configurations with multiple peers which have similar settings, create a *BFD profile* with the common settings before creating peer entries.

BFD **Peers** are managed at **Services > FRR BFD** on the **Peers** tab.

The **Peers** tab contains a list of current BFD peers, if any, and controls to manage the entries (e.g. edit, delete). The  **Add** button creates a new peer.

The remaining sections on this page cover the various options available when creating or editing a peer entry.

Peer Configuration

Peer Address The remote BFD peer address. The local and remote peer IP addresses must use the same address family (either IPv4 or IPv6)

Description A text description of this BFD peer.

Profile A *BFD Profile* from which settings will be used, unless overridden on this entry.

Options

Multihop Expect packets with a TTL value less than `254` indicating there is more than one hop between peer addresses. Also listens on the multihop port, `4784`.

In most cases this option is not necessary, but certain configurations may require it, such as running BGP on loopback interfaces instead of an interface directly connected to a peer.

Warning: When using multi-hop mode echo-mode will not work, see RFC 5883 section 3.

Shutdown Enables or disables the peer administratively.

When the peer is disabled in this way an "administrative down" message is sent to the remote peer.

Source Address/Interface

Interface The interface on which to enable BFD.

Local Source Address The local address used as a source for BFD packets, if it differs from the primary IP address on **Interface**.

Advanced Options

Detect Multiplier A non-zero value that is, roughly speaking, due to jitter, the number of packets that have to be missed in a row to declare the session to be down. Must be between `2` and `255`.

The remote transmission interval will be multiplied by this value to determine the connection loss detection timer. The default value is `3`.

Receive Interval The minimum interval, in milliseconds, at which this router can receive control packets from a peer. The default value is `300`.

Transmit Interval The minimum transmission interval, in milliseconds, for the router to send BFD control packets to peers. The default value is `300`.

Echo Interval The minimum interval, in milliseconds, at which this router can receive echos from a peer. The default value is `50`.

Echo Mode Enables or disables echo transmission mode. By default, this mode is disabled.

> **Warning:** Echo mode is not supported on multi-hop setups, see RFC 5883 section 3.

> **Tip:** FRR documentation recommends that the transmission interval of control packets to be increased after enabling echo mode to reduce bandwidth usage.
>
> For example, increase **Transmit Interval** from `300` to `2000`.

BFD Profiles

BFD profiles hold common sets of BFD settings which can be shared my multiple BFD peers. This can speed up the configuration process when there are numerous similar peers.

BFD **Profiles** are managed at **Services > FRR BFD** on the **Profiles** tab.

The **Profiles** tab contains a list of current BFD profiles, if any, and controls to manage the entries (e.g. edit, delete).

The  **Add** button creates a new profile.

The remaining sections on this page cover the various options available when creating or editing a profile entry.

Note: Most of the options in BFD profiles are the same as those for *BFD Peers*. This document only covers the options which are set on profiles only, not those shared with peers.

Profile Configuration

Name The name of this profile. This name appears in the drop-down list for **Profile** on the *BFD Peers* configuration.

Passive Marks sessions using this profile as passive.

A passive session will not attempt to start the connection and will wait for control packets from peer before it begins replying.

Profile Options

Minimum TTL For multihop sessions only, configure a specific minimum expected TTL value for incoming BFD control packets.

This feature tightens the packet validation requirements to avoid receiving BFD control packets from other sessions.

The default value is `254` which means there is only one hop between this router and the peer.

Using BFD

For BFD to function fully, the BFD session status must be consumed by other interested parties. Currently this can be BGP, OSPF, or OSPF6.

BGP BFD can be enabled for specific BGP neighbors using the *BFD options for BGP neighbors*.

OSPF/OSPF6 BFD can be enabled for OSPF and OSPF6 by using the appropriate *OSPF interface BFD option* or *OSPF6 interface BFD option*.

29.10.6 Raw FRR Configurations

Rather than using the GUI, administrators can opt to manually manage the FRR configuration. This allows for more complex configurations than can be supported in the GUI, but is also more prone to error and less capable of acting dynamically based on certain factors such as changing interface addresses.

Manual TCP MD5 Peers

When using TCP MD5 protection on BGP, this option defines MD5 passwords used to communicate with specific remote peers. This must be defined separately as it requires special handling in the operating system outside of FRR.

Raw Configuration Management

The **Update Running** button reads the current FRR configuration into the **Running frr.conf** field on this page.

Raw Configuration Files

Saved frr.conf This is the FRR configuration which will be loaded by FRR when starting up.

Running frr.conf This is the configuration data currently active in FRR. Editing the running configuration will result in changes to the active FRR configuration, but is not stored long term.

To save the configuration for later use (e.g. when booting up or restarting FRR), click the **Copy frr.conf Running to Saved** button.

To refresh this based on the current information from the running instance of FRR, click **Update Running** in the **Raw Configuration Management** section.

29.10.7 Dynamic Routing Protocol Lists

Throughout the FRR package, certain options specify a supported routing protocol or source of routes. Currently, the following values can be found in these locations, but not every option appears in each area:

connected Routes for directly connected networks on up and active interfaces.

kernel Routes from the kernel, including static routes defined outside of FRR and other non-dynamic routes.

FRR Static Static routes defined in the FRR configuration

bgp Routes obtained dynamically from BGP neighbors

ospf IPv4 routes obtained dynamically from OSPF neighbors

ospf6 IPv6 routes obtained dynamically from OSPF6 neighbors

29.11 HAProxy package

HAProxy is a free, very fast and reliable solution offering high availability, load balancing, and proxying for TCP, HTTP and HTTPS-based applications. It is particularly suited for web sites struggling under very high loads while needing persistence or Layer7 processing.

Supporting tens of thousands of connections is clearly realistic with todays hardware. Its mode of operation makes integrating it into existing architectures very easy and riskless, while still offering the possibility not to expose fragile web servers to the Net.

Refer to the following articles for more information on the listed topics:

See also:

- *Troubleshooting the HAProxy Package*

29.11.1 Package Variants

On recent pfSense® versions 2 haproxy packages are available:

- **HAProxy** package tracks the stable FreeBSD port currently using HAProxy 1.6.x.
- **HAProxy-devel** package uses haproxy-devel from FreeBSD ports and loosely tracks HAProxy 1.7dev new features in the pfSense package are also first included in the HAProxy-devel then later copied over the HAProxy package.

For info about HAProxy 1.6 and 1.7 see: https://github.com/PiBa-NL/pfsense-haproxy-package-doc/wiki

29.11.2 Recent Changes

See github log for recent changes:

- https://github.com/pfsense/FreeBSD-ports/commits/devel/net/pfSense-pkg-haproxy
- https://github.com/pfsense/FreeBSD-ports/commits/devel/net/pfSense-pkg-haproxy-devel

Older Releases:

- version as of July 7, 2013: 1.4.24 pkg v 1.2
- 1.00 - A lot of changes and fixes:
  - **Versioning changed** - The package will show the version as "1.4.18 pkg v 1.0" where the first version is the version of the HAProxy binary, and the second is the package version.
  - HAProxy has been updated to 1.4.18, with separate compiled versions for 32 bit (i386) and 64 bit (amd64).
  - **New Status Options** - In addition to *active* and *inactive*, the options *backup* and *disabled* are available as a status for servers. These options translate directly to the equivalent options in HAProxy.
  - **Monitor URI is now Optional** - Having monitor URI as a required option prevented non-HTTP TCP frontends from operating properly.
  - **XMLRPC Sync** - Fixed two problem that prevented configuration sync from working correctly.
  - **New Stats Options** - Node name, node description, and auto-refresh are now available for the stats page.
  - **New Load Balancing Options** - The *static-rr* and *leastconn* load balancing options offered by HAProxy are not available options when creating frontends.
  - **Ports Text Box** - The ports box takes a comma separated list of ports but was limited to 10 characters. This has been raised to 500.
  - **Global Advanced Options** - Fixed a problem with the *Advanced Options* box on the globals tab that caused it to garble the text in both the HAProxy config and on display in the UI.
  - **Default Tab** - The default tab has been changed to the global tab rather than the frontends tab. Makes it easier to see the generated config after saving.
- 0.29 - option to output the automatically generated haproxy.cfg on the global settings tab.
- 0.28 - nbproc is now a setting on the global settings tab (number of processes in haproxy manual speak). local0 is now appended to the ip address of the remote syslog server.
- 0.27 - open files limit increased, check inter now an option. http close and forward for now working correctly when unchecked
- 0.26 - haproxy application version bumped to HA-Proxy version 1.3.22 2009/10/14
- 0.25 - advanced box now allows carriage returns to separate multiple options/directives
- 0.24 - Retries is no longer a required field
- 0.23 - Advanced box added which allows passing thru to the haproxy configuration file options or directives that are not supported by the GUI.
- 0.22 - httpclose is now an option
- 0.21 - forwardfor is now an option
- 0.19 - Remote logging feature added
- 0.18 - Ability to select multiple frontends added
- 0.16 - https mode added which will automatically adjust backend configuration

- 0.15 - Configuration synchronization added for CARP clusters
- 0.01 - Converted to a pfSense package

29.11.3 Known issues

- When editing a frontend, choose an **External Address** and the description refers to "the interface chosen above" but it isn't possible to choose an interface. The only options for the address are *Interface Address*, *Any*, and any defined VIPs. I assume that the interface it chooses normally is the WAN, but this should be clarified.
- Selecting Cancel when editing a frontend or server seems to use browser back; that means if returning to the previous page from a POST, it will ask to resubmit the data, which is not desirable.
- When changes are applied, it appears that everything was successful whether it is or not. If something caused an invalid HAProxy config to be generated, HAProxy will not be stopped nor will it be restarted. There is no error message to indicate that this is the case and the **Apply Changes** button will disappear.
- Most time-based options appear to be optional and specify "default" amounts but are actually required (and shouldn't be).
- Some unexpected behavior for those who are used to HAProxy configuration alone (see the differences section).

HAProxy has their own list of known bugs by branch and by version.

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.11.4 Things that could be improved

- Configuration Sync should have X number of possible nodes, in the form of a list where additional nodes are added, rather than 3 static boxes.
- An overhaul of the package to fully utilize HAProxy's capabilities. It would be difficult to simply add in tabs for the other sections due to internal naming. In the package, the "frontends" set up in the Frontends tab are referred to in the configuration as backends. Including full support for frontends, backends, and listen sections would make it possible to use a lot more advanced features even without direct GUI support for them (through the use of the advanced pass thru box).
- HAProxy has options for soft-restarts, which are useful in a high-availability environment. Some changes would be needed to the way the **Apply Changes** button works, or maybe make it optional. It needs more research.
- Make all time fields accept the same time units that HAProxy does for the particular field.

29.11.5 Differences between this package and HAProxy used directly

HAProxy defines five main sections in its configuration.

- global
- defaults
- frontend
- backend
- listen

global defines options that process-wide and often OS-specific.

defaults sets default parameters for all other sections following its declaration.

frontend describes a set of listening sockets accepting client connections.

backend describes a set of servers to which the proxy will connect to forward incoming connections.

listen defines a complete proxy with its frontend and backend parts combined in one section. It is generally useful for TCP-only traffic.

In the pfSense package, tabs exist to define "frontends" and "servers" but the resulting configuration is actually made up completely of listen sections. This is okay for the most part, but it does prevent advanced usages that need to refer to several backends and the like.

In HAProxy, a single server directive can be made with a blank port and it will listen on all the ports of the frontend that it is assigned to. The package's GUI implies that this will be the case by leaving the port blank.

What actually gets generated instead is a single server directive for each port that the frontend is listening on. This is an important difference when the ports that are being listened on are not interchangeable. Example:

Define a front end for SMTP connections listening on ports 25 and 465. The server is listening on both of those ports, but 25 does not accept SSL/TLS and 465 does. When someone connects to the proxy on port 25, they should get connected to the server on port 25, and when they connect on 465, they get connected to the server on port 465.

In a standard HAProxy configuration where the frontend is set to listen on both ports and a single server directive is made with no port, it will operate the expected way.

In pfSense software, two server directives will be generated; one for each port. HAProxy will not send connections the expected way. It will loadbalance between them, regardless of whether the frontend and server ports match.

Therefore in pfSense software a separate frontend must be created for this, as they are essentially different services. Listen on port 25 and 2525, and it doesn't matter whether someone connected on one port gets directed to the other, then they can be combined.

Splitting the servers up by port also means that a separate entry will exist for each one in the stats page, but the port will not be shown. In an HAProxy configuration where a single server directive has no ports and effectively handles multiple (due to inheriting from the frontend) it will only show up in the stats once.

29.11.6 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.12 iperf package

iperf is a tool used for network throughput testing.

29.12.1 Usage

iperf running on pfSense® software is NOT a suitable way of testing firewall throughput, as there is a significant difference between performance of traffic *initiated or terminated on the firewall* and traffic *traversing the firewall*. There are many suitable uses for iperf running on pfSense software, but testing the throughput capabilities of the firewall is not one of them.

Uses for iperf on pfSense software

- Measuring throughput from the internal network to the inside of the firewall
  - Useful in scenarios where portions of the internal network are behind links slower than the firewall's network interface. Examples include testing the throughput of a wireless network or private WAN network connected to a router inside the network.
- Testing end to end throughput between two firewalls on the Internet
- Anything else where performance measurement excluding throughput capabilities of the firewall are desirable.

29.12.2 Additional Resources

- iperf Man Page
- iperf - The Easy Tutorial

29.12.3 Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.12.4 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.13 IPsec Export Package

The IPsec Export package generates client configurations for mobile IPsec, making it easier to configure remote access clients. This package is available on pfSense® Plus as well as older Factory Edition versions of pfSense software.

The IPsec Export package contains an IPsec Profile export page for Apple devices and an IPsec Export page for Windows. Both pages work in a similar manner, and give administrators a few extra options to control client behavior.

The package works with most types of mobile IPsec configurations, with some exceptions depending upon settings.

This utility checks configured Mobile Phase 1 and Phase 2 entries and attempts to locate a set of parameters which are compatible with clients. It uses the first match it finds, so order choices in the Phase 1 and Phase 2 list appropriately or manually edit the resulting profile or script as needed.

Note: Apple and Windows do not support certain settings. In these cases, the package will print a notice that it is not possible to export a configuration.

See the Apple Configuration Profile Reference Documentation for details about the contents of profiles and settings they support.

For a full list of parameters compatible with Windows clients, see the Microsoft Documentation for `Set-VpnConnectionIPsecConfiguration`.

29.13.1 Export Settings

When exporting an IPsec configuration, the following options are available to fine-tune the values put into the generated configuration.

VPN Name The name of the VPN as seen by the client in their network list. This name is also used when creating the filename of the files exported by the package. It is pre-filled with some basic information, such as the firewall hostname, but it can be customized.

Server Address Select the server address to be used by the client. This list is generated from the SAN entries on the server certificate.

The hostname used by the client to connect to the server must exist in DNS and it must be present in the server certificate SAN list for the client to properly validate the certificate.

Set to *Custom Hostname* to fill in a hostname other than one shown in the list.

Custom Hostname A text field for a custom fully qualified domain name to which the client can connect. As with the **Server Address**, this must exist in DNS and be in the server certificate SAN list.

VPN Client (Apple) The user for which the package will generate a configuration. Depending on the Mobile IPsec Phase 1 settings, this could either be a user or a TLS certificate.

When using certificates, the list contains certificates which were signed by the CA selected on the mobile IPsec Phase 1 **IPsec Peer Certificate Authority**.

Note: This differs from Windows because Windows can prompt for a username, but Apple requires it to be present in the profile.

TLS User Certificate (Windows) The TLS user certificate to include in the exported configuration, if needed. This field is only visible when the Mobile IPsec Phase 1 settings require a client certificate (e.g. EAP-TLS).

29.13.2 Export a Client Configuration

The process to export a client for an existing *Mobile IPsec* configuration varies slightly for Apple and Windows.

Apple

- Navigate to **VPN > IPsec Export: Apple**
- Configure the settings as described in *Export Settings*
- Click **View** to display the generated configuration profile
- Review the profile contents and confirm it is acceptable
- Click **Download** to download the configuration profile

Apple Client Configuration

Visit the Apple Configurator Site for details about creating and using profiles. The process varies between iOS and OS X.

Windows

- Navigate to **VPN > IPsec Export: Windows**
- Configure the settings as described in *Export Settings*
- Click **View** to display the generated PowerShell script
- Review the script contents and confirm it is acceptable
- Click **Download** to download a ZIP archive containing the PowerShell script and the required certificates.

If the **Network List** option is active on the **Mobile Clients** tab in IPsec settings, the script will include parameters to setup Split Tunneling on the client as well as commands to configure routes on the VPN for networks configured in the mobile Phase 2 entries.

Windows Client Configuration

On the client system, unzip the configuration archive and run the script. The commands in the PowerShell script will import certificates and setup the VPN on the client workstation.

Running PowerShell scripts on Windows is disabled by default. If scripting is disabled, the commands may be copied and pasted into a PowerShell prompt.

See also:

Local policies may override that behavior. See the PowerShell Execution Policies Documentation for details.

> **Warning:** Some commands may require Administrator access, such as importing the CA certificate. Run these commands at an Administrator-level PowerShell prompt or use an alternate method.

29.14 Using LLDP on pfSense software

Note: Use the **LADVD** package available through the pfSense® webGUI.

29.14.1 lldpd

lldpd allows LLDP to be enabled on pfSense software.

Install

- Navigate to **System > Packages**, **Available Packages** tab
- Search for `lldp` or find it in the list
- Click **Install** and **Confirm**

The package adds a menu entry under **Services > LLDP**.

Setup

The settings are under **Services > LLDP**, **LLDP Settings** tab.

At a minimum, check **Enable**, pick **Interfaces**, and then set other options as desired.

Seeing neighbors

Navigate to **Services > LLDP**, **LLDP Status** tab. The neighbors will be printed in the lower box.

29.15 Nmap package

nmap is a powerful network scanner that provides port scanning, OS and service identification, and more.

29.15.1 Usage

After installing the package, `nmap` will be available at **Diagnostics > nmap** as well as in the shell (SSH or Console).

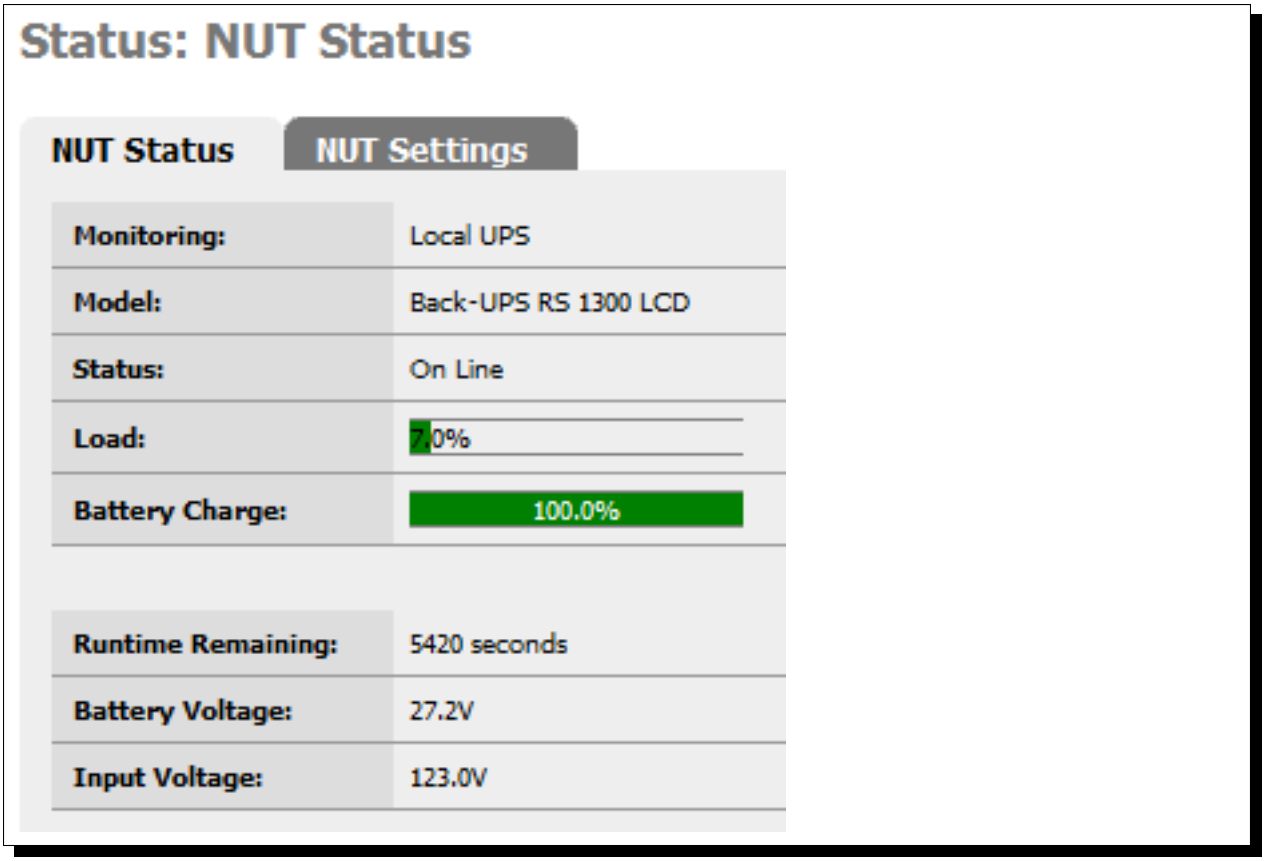
The nmap package also installs an OUI database that is used by the pfSense® webGUI to display manufacturer names on pages that list MAC addresses, such as the *ARP Table*, and *DHCP Leases*.

29.15.2 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.16 Nut package

NUT status screen



The NUT package provides a way to monitor an Uninterruptable Power Supply (UPS) using Network UPS Tools (NUT).

After installation, the package may be configured at **Services > UPS**.

See also:

Visit the NUT Package forum thread for assistance with this package.

29.16.1 Troubleshooting

The package does not have a lot of input validation in the webGUI. If NUT will not start after configuration, check the System log from the pfSense® webGUI from **Status > System Logs**) for log entries starting with `nut:`. The culprit is likely explained there, such as selecting a cable for a driver type that does not need (nor permit) the cable selection.

If the system log doesn't offer adequate information, such as simply logging:

```
nut: Service failed to start: check configuration
```

Log in via SSH and choose option `8`, then run the following command:

```
/usr/local/etc/rc.d/nut.sh start
```

Configuration errors will be displayed in the output if any are found.

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.17 Open VM Tools package

This package installs VMware Tools for pfSense® software, using the Open VM Tools available from VMware. It is built using the open-vm-tools-nox11 FreeBSD port.

29.17.1 Usage

There is no web GUI, the services are automatically started at boot time.

29.17.2 Verifying functionality

There are two portions of this package, the **vmware-guestd** process and the kernel modules.

Verifying vmtoolsd

Navigate to **Diagnostics > Command**, and in **Execute shell command**, enter the following:

```
ps uxawww | grep vmtoolsd
```

Output similar to the following will be shown:

```
$ ps uxawww | grep vmtoolsd
1026  ??  Ss     0:20.84 /usr/local/bin/vmtoolsd -c /usr/local/share/vmware-tools/
↪tools.conf -p /usr/local/lib/open-vm-tools/plugins/vmsvc
19157  ??  SN     0:00.00 sh -c ps ax|grep vmware
19159  ??  RN     0:00.00 grep vmware (sh)
```

As long as **vmtoolsd** is shown in the output, it is working.

29.17.3 Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.17.4 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.18 OpenVPN Client Export Package

The easiest way to configure an OpenVPN client on most platforms is to use the OpenVPN Client Export Package on the pfSense® firewall.

Install the OpenVPN Client Export Utility package as follows:

- Navigate to **System > Packages**
- Locate the **OpenVPN Client Export** package in the list

- Click  **Install** next to that package listing to install

Once installed, it can be found at **VPN > OpenVPN**, on the **Client Export** tab.

The options for the package include:

Remote Access Server Pick the OpenVPN server instance for which a client will be exported. If there is only one OpenVPN remote access server there will only be one choice in the list. The list will be empty if there are no Remote Access mode OpenVPN servers.

Host Name Resolution Controls how the "remote" entry the client is formatted.

Interface IP Address When chosen, the interface IP address is used directly. This is typically the best choice for installations with a static IP address on WAN.

Automagic Multi-WAN IPs This option is useful when redirecting multiple ports using port forwards for deployments that utilize multi-WAN or multiple ports on the same WAN. It will seek out and make entries for all port forwards that target the server and use the destination IP address used on the port forward in the client configuration.

Automagic Multi-WAN DDNS Hostnames Similar to the previous option, but it uses the first Dynamic DNS entry it finds that matches the chosen destination.

Installation Hostname Places the firewall's hostname, defined under **System > General Setup**, into the client configuration. The hostname must exist in public DNS so it can be resolved by clients.

Dynamic DNS Hostname Entries Each Dynamic DNS hostname configured on the firewall is listed here. These are typically the best choice for running a server on a single WAN with a dynamic IP address.

Other Presents a text box in which a hostname or IP address can be entered for the client to use.

Verify Server CN Specifies how the client will verify the identity of the server certificate. The CN of the server certificate is placed in the client configuration, so that if another valid certificate pretends to be the server with a different CN, it will not match and the client will refuse to connect.

Automatic - Use verify-x509-name where possible This is the best for current clients. Older methods have been deprecated since this method is more accurate and flexible.

Use tls-remote This can work on older clients (OpenVPN 2.2.x or earlier) but it will break newer clients as the option has been deprecated.

Use tls-remote and quote the server CN Works the same as *tls-remote* but adds quotes around the CN to help some clients cope with spaces in the CN.

Do not verify the server CN Disables client verification of the server certificate common name.

Use Random Local Port For current clients, the default (checked) is best, otherwise two OpenVPN connections cannot be run simultaneously on the client device. Some older clients do not support this, however.

Use Microsoft Certificate Storage Under Certificate Export Options, for exported installer clients this will place the CA and user certificate in Microsoft's certificate storage rather than using the files directly.

Use a password to protect the pkcs12 file contents When checked, enter a Password and confirm it, then the certificates and keys supplied to the client will be protected with a password. If the OpenVPN server is configured for user authentication this will cause users to see two different password

prompts when loading the client: One to decrypt the keys and certificates, and another for the server's user authentication upon connecting.

Use Proxy If the client will be located behind a proxy, check *Use proxy to communicate with the server* and then supply a Proxy **Type**, **IP Address**, **Port**, and **Proxy Authentication** with credentials if needed.

OpenVPNManager When checked, this option will bundle the Windows installer with OpenVPNManager GUI in addition to the normal Windows client. This alternate GUI manages the OpenVPN service in such a way that it does not require administrator-level privileges once installed.

Additional configuration options Any extra configuration options needed for the client may be placed in this entry box. This is roughly equivalent to the **Advanced options** box on the OpenVPN configuration screens, but from the perspective of the client.

Note: There is no mechanism to save these settings, so they must be checked and set each time the page is visited.

29.18.1 Client Install Packages List

Under **Client Install Packages** is a list of potential clients to export. The contents of the list depend on how the server is configured and which users and certificates are present on the firewall.

The following list describes how the server configuration style affects the list in the package:

Remote Access (SSL/TLS) User certificates are listed which are made from the same CA as the OpenVPN server

Remote Access (SSL/TLS + User Auth – Local Users) User entries are listed for local users which also have an associated certificate made from the same CA as the OpenVPN server.

Remote Access (SSL/TLS + User Auth – Remote Authentication) Because the users are remote, user certificates are listed which are made from the same CA as the OpenVPN server. It is assumed that the username is the same as the common name of the certificate.

Remote Access (User Auth – Local Users or Remote Authentication) A single configuration entry is shown for all users since there are no per-user certificates.

The example setup from the wizard made previously in this chapter was for SSL/TLS + User Auth with Local Users, so one entry is shown per user on the system which has a certificate created from the same CA as the OpenVPN server.

Note: If no users are shown, or if a specific user is missing from the list, the user does not exist or the user does not have an appropriate certificate. See *Local Users* for the correct procedure to create a user and certificate.

29.18.2 Client Install Package Types

Numerous options are listed for each client that export the configuration and associated files in different ways. Each one accommodates a different potential client type.

Standard Configurations

Archive Downloads a ZIP archive containing the configuration file, the server's TLS key if defined, and a PKCS#12 file which contains the CA certificate, client key, and client certificate. This option is usable with Linux clients or Tunnelblick, among others.

File Only Downloads only the basic configuration file, no certificates or keys. This would mainly be used to see the configuration file itself without downloading the other information.

Inline Configurations

This choice downloads a single configuration file with the certificates and keys inline. This format is ideal for use on all platforms, especially Android and iOS clients or for manually copying a configuration to a system that already has a client installed. This option will work for any client type based on OpenVPN version 2.1 or newer.

Android Used with the Android OpenVPN client mentioned in *Installing the OpenVPN Client on Android*.

OpenVPN Connect (iOS/Android) Used with the OpenVPN Connect client on iOS or Android described in *Installing the OpenVPN Client on iOS*.

Others Usable by any standard OpenVPN client on platforms such as Windows, OS X, or BSD/Linux. It also works well with Tunnelblick on OS X, simply download the inline config and drag it into the configurations folder for Tunnelblick.

SIP Phone archives

If the OpenVPN server is configured as SSL/TLS only *without* authentication then options will appear to export client configurations for several models of SIP handsets that support OpenVPN. Notable examples are the Yealink T28 and T38G, and SNOM phones. Installing the client to the phone varies by model, check the manufacturer's documentation for more information.

Note: Ensure the phone has a proper clock setup and/or NTP server, otherwise the certificates will fail to validate and the VPN will not connect.

Warning: Typically these handsets only support the use of SHA1 as a certificate hash. Ensure the CA, server certificate, and client certificates are all generated using *SHA1* or they may fail. They may also only support a limited set of encryption algorithms such as AES-128-CBC. Consult the phone documentation for details.

Windows Installers

The Windows Installer options create a simple-to-use executable installer file which contains the OpenVPN client with the configuration data embedded. The installer runs like the normal Windows OpenVPN client installer, but it also copies all of the settings and certificates needed. See *Installing the OpenVPN Client on Windows* below for some notes on how to install and run the Windows client.

Currently, there are four options available:

x86-xp 32-bit installer usable on Windows XP and later

x64-xp 64-bit installer usable on Windows XP and later

x86-win6 32-bit installer usable on Windows Vista and later and includes a newer tap driver

x64-win6 64-bit installer usable on Windows Vista and later and includes a newer tap driver

Note: Be sure to click next/finish all the way through the installation process. Do not click cancel or X out the install at any step, or the client system may be left with the client installed but no imported configuration.

Warning: On Windows Vista, 7, 8, 10 and later with UAC (User Account Control) enabled, the client **must** be run as Administrator. Right click the OpenVPN GUI icon and click **Run as Administrator** for it to work. It can connect without administrative rights, but it cannot add the route needed to direct traffic over the OpenVPN connection, leaving it unusable. The properties of the shortcut may be set to always launch the program as Administrator. This option is found on the **Compatibility** tab of the shortcut properties. One way around that requirement is to check **OpenVPNManager** before exporting to use an alternate OpenVPN management GUI on Windows.

The Viscosity client is also available for Windows and it does not require administrative privileges to run properly.

Viscosity Bundle

This works like the configuration archive above, but is for the Viscosity OpenVPN client used in OS X and Windows. If the Viscosity client is already installed, download this bundle and click it to import it into the client.

29.19 OpenVPN Client Import Package

Note: This package is only available on Netgate pfSense® Plus software.

The OpenVPN client import package can take a unified OpenVPN client configuration file as exported by an OpenVPN server and automatically turn it into an OpenVPN client instance on pfSense Plus software. The unified OpenVPN configuration file format includes all of the certificates and keys required for the connection, allowing the client instance to be created with minimal effort.

In many cases the newly imported client instance starts and passes traffic on completion of the import, but in some cases adjustments must be made to the imported client configuration by editing the resulting OpenVPN client instance.

The package can be installed using the *Package Manager* on a pfSense Plus installation. Once the package is installed, it can be accessed at **VPN > OpenVPN** on the **Import** tab.

29.19.1 How it Works

The import process attempts to read the configuration file and map directives from the file to their equivalent settings in pfSense Plus software. Unknown directives are placed into the **Custom options** area in the resulting client instance.

If the configuration being imported contains certificates, the import package will create appropriate CA and certificate entries if they do not already exist.

Note: If the configuration requires certificates but they are not present in the imported configuration file, they can be manually imported in the certificate manager and then manually selected in the OpenVPN client instance after it has been imported.

Once the import process is complete, the new client is stored and, if it is enabled and has a complete configuration, the client is immediately started.

29.19.2 Imported OpenVPN Client Configuration

When importing a configuration there are several options specific to pfSense Plus software which cannot be automatically determined from the imported configuration. These must be filled in manually before the import process can be completed.

These options are equivalent to their counterparts in the *OpenVPN Configuration Options*. Consult that document for additional details on these settings.

Config File The OpenVPN configuration file (e.g. `<name>.ovpn`) to import.

The OpenVPN client configuration file can be from another instance of pfSense software, a VPN provider, or other OpenVPN compatible server so long as it uses the standard OpenVPN configuration format.

Disabled When set, the client will be marked as disabled on import so it will not start automatically.

Server Mode Chooses between whether this client is connecting to an SSL/TLS server with certificates, or to a shared key server.

Name A descriptive name for this client instance.

Interface The firewall interface to be used by this client instance for outbound connections. In most cases this will be **WAN** but may also be another interface, or a virtual IP address.

Username The username to use if the OpenVPN server requires a username and password. May be left blank if the server does not require user authentication.

Password The password to use if the OpenVPN server requires a username and password. May be left blank if the server does not require user authentication.

29.19.3 Client Import Example

The process to import a client generally follows this format:

- Obtain an OpenVPN configuration file in inline format from the OpenVPN server (e.g. `username.ovpn`)

Note: If the server is also running pfSense software, use the *OpenVPN Client Export Package* and download the inline configuration using the **Most Clients** button.

- Navigate to **VPN > OpenVPN**, **Import** tab on the client firewall
- Click **Browse** in the **.ovpn config file** field and select the configuration file obtained from the server (e.g. `username.ovpn`)
- Fill in the other options as described in *Imported OpenVPN Client Configuration*
- Click **Import**

At that point the client instance will be created and started automatically. If the configuration was incomplete or needs other changes, then do so as follows:

- Navigate to **VPN > OpenVPN**, **Clients** tab
- Find the newly imported client in the list and click  on its row

- Make final adjustments needed
- Click **Save**

See also:

See also: *OpenVPN Configuration Options*

29.20 pfBlocker-NG Package

pfBlocker-NG introduces an Enhanced Alias Table Feature to pfSense® software.

What it allows:

- Assigning many IP address URL lists from sites like I-blocklist to a single alias and then choose a rule action.
- Blocking countries and IP ranges.
- Replacement of both **Countryblock** and **IPblocklist** by providing the same functionality, and more, in one package.
- Uses native functions of pfSense software instead of file hacks and table manipulation.

Features include:

- Country_Block features
- IP_Blocklist features
- Dashboard widget
- XMLRPC Sync
- Dashboard widget with aliases applied and package hit
- Lists update frequency
- Many new options to choose what to block and how to block.
- Network lists may be used for custom rules.

29.20.1 General Setup

Set the interfaces to be monitored by pfBlocker-NG (both inbound and outbound), where the inbound is the Internet connection.

To prevent devices or users from accessing sites in the selected countries/IP addresses, select local interfaces under **outbound**.

29.20.2 Setting up Lists

This is the IPBlocklist feature, enter IP addresses here to specifically block. It must be in the file format or CIDR. Create a list for each type of action to be taken by pfBlocker.

Options are:

`Deny Both` - Will deny access on Both directions.

`Deny Inbound` - Will deny access from selected lists to the local network.

`Deny Outbound` - Will deny access from local users to IP address lists selected to block.

`Permit Inbound` - Will allow access from selected lists to the local network.

`Permit Outbound` - Will allow access from local users to IP address lists selected to block.

`Disabled` - Will just keep selection and do nothing to selected Lists.

`Alias Only` - Will create an alias with selected Lists to help custom rule assignments.

The rest of the tabs (except sync) specify the other lists included with the package. They are separated by continent with the exception of the spammer list which contains countries from around the globe that are known to harbor spammers.

Sync tab configures pfBlocker to sync its configuration to other pfSense devices.

29.20.3 Available lists

Spamhaus - DROP and EDROP.

- http://www.spamhaus.org/drop/drop.txt
- http://www.spamhaus.org/drop/edrop.txt

DShield - Most Active Attacking IPs.

- http://feeds.dshield.org/top10-2.txt

iblocklist.com - A number of lists are available.

- http://www.iblocklist.com/lists.php

29.20.4 FAQ

I'm getting memory errors while applying pfblocker lists, how to fix this? Increase table size to avoid memory errors in Advanced settings.

I can't see any pfblocker rules applied, whats wrong? pfblocker requires at least one firewall entry (any interface) for it to be active. One way to verify is to check the front page widget.

pfBlocker always moves its rules to the top, how can I stop this? Change rule action to Alias only and then apply custom rules using pfBlocker aliases with an arbitrary sequence.

How can I apply pfBlocker lists in floating rules? Aliases are used for customized filter entries and float rules.

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.21 Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) daemon is a dynamic routing protocol which, when coupled with other routers that also have RIP enabled, will allow automatic route updates between them.

To enable RIP on a pfSense® router, install the package and then navigate to **Services > RIP**. Check the box to **enable** the service, ctrl-click to select **Interfaces** to which RIP will bind, select a **RIP version**, and enter a **RIPv2 password** if using RIP version 2.

Warning: This package has been deprecated.

29.22 Siproxd package

Warning: For many modern SIP configurations this package is unnecessary. Please only install and use siproxd if it is absolutely required.

Siproxd is a proxy/masquerading daemon for SIP. It handles registrations of SIP clients on a private IP network and performs rewriting of the SIP message bodies to make SIP connections possible via a masquerading firewall. It allows SIP phones and soft phones (like kphone, linphone) to work behind an IP masquerading firewall or router.

The most useful thing siproxd does is allow multiple phones to use a static source port of 5060 when registering to the same remote PBX. **If the PBX is local, this package should not be installed or used**. Most remote PBX systems are OK with the phones having random source ports, but that was not the case many years ago. Unless the remote PBX is absolutely strict about the 5060 source port requirement for each phone, this package is not needed.

29.22.1 Install siproxid

In the pfSense® webGUI, navigate to **System > Packages**:

- Find the **siproxd** package.
- Click  to install, and confirm installation.
- Wait for it finish.

29.22.2 Configure siproxd

Under **Services > siproxd**:

- **Inbound interface** will generally be `LAN`.
- **Outbound interface** will generally by `WAN`.
- Fill in any other values (where appropriate).
- **RTP port range (lower)** should be an even number.
- Click **Save**.

29.22.3 Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.22.4 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.23 IDS / IPS

pfSense® software can act in an Intrusion Detection System (IDS) / Intrusion Prevention System (IPS) role with add-on packages like Snort and Suricata.

Note: The Snort and Suricata packages share many design similarities, so in most cases the instructions for Snort carry over to Suricata with only minor adjustments.

29.23.1 Snort

Configuring the Snort Package

Snort is an intrusion detection and prevention system. It can be configured to simply log detected network events to both log and block them. Thanks to OpenAppID detectors and rules, Snort package enables application detection and filtering. The package is available to install in the pfSense® webGUI from **System > Package Manager**. Snort operates using detection signatures called rules. Snort rules can be custom created by the user, or any of several pre-packaged rule sets can be enabled and downloaded.

The Snort package currently offers support for these pre-packaged rules:

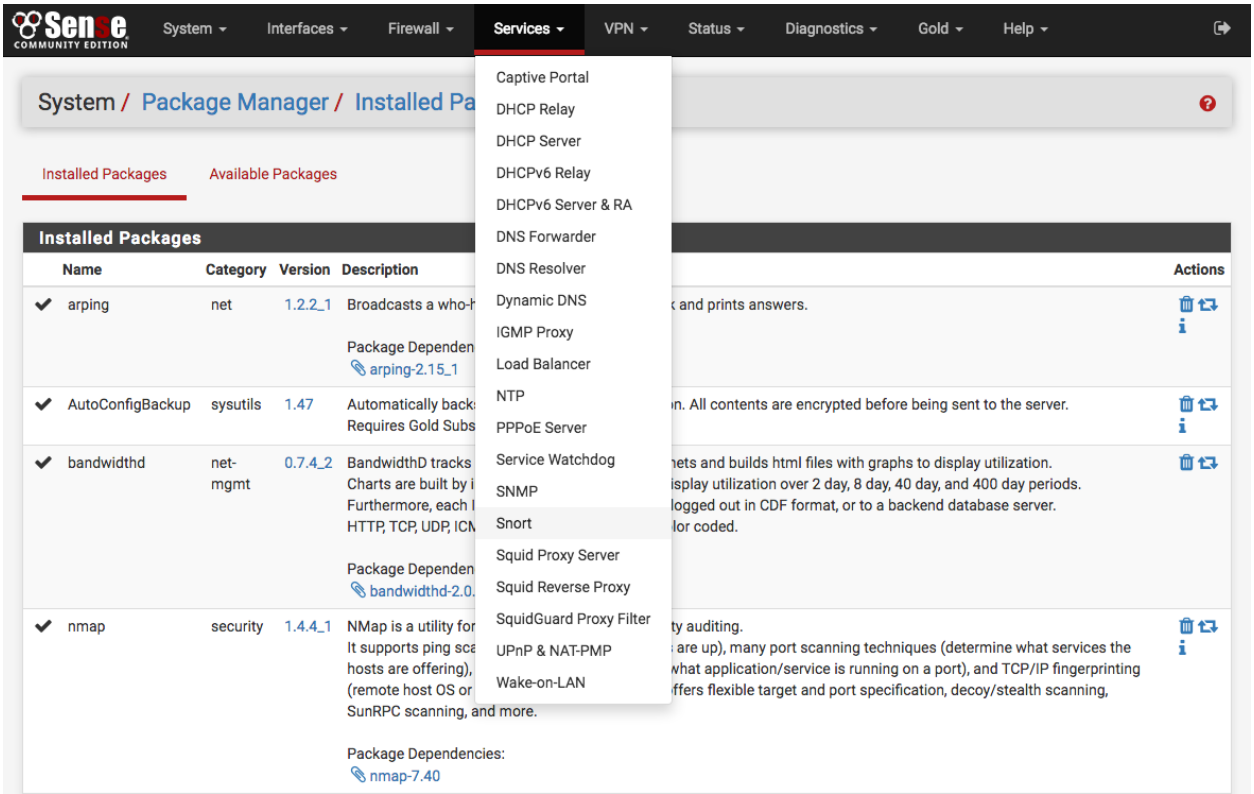
- Snort VRT (Vulnerability Research Team) rules
- Snort GPLv2 Community Rules
- Emerging Threats Open Rules
- Emerging Threats Pro Rules
- OpenAppID Open detectors and rules for application detection

The Snort GPLv2 Community Rules and the Emerging Threats Open Rules are both available for free with no registration required. The Snort VRT rules are offered in two forms. One is a registered-user version which is free, but requires registration at http://www.snort.org. The registered-user free version only provides access to rules that are 30-days old or more in age. A Snort VRT paid subscription can be purchased, and it offers twice-weekly (and sometimes more frequent) updates to the rules. The Emerging Threats Pro rules are offered to paid subscribers only and offer almost daily updates to address fast-changing threats.

We strongly suggest obtaining a paid subscription from Snort or Emerging Threats in order to download the most current rules. This is highly recommended for commercial applications.

Launching Snort configuration GUI

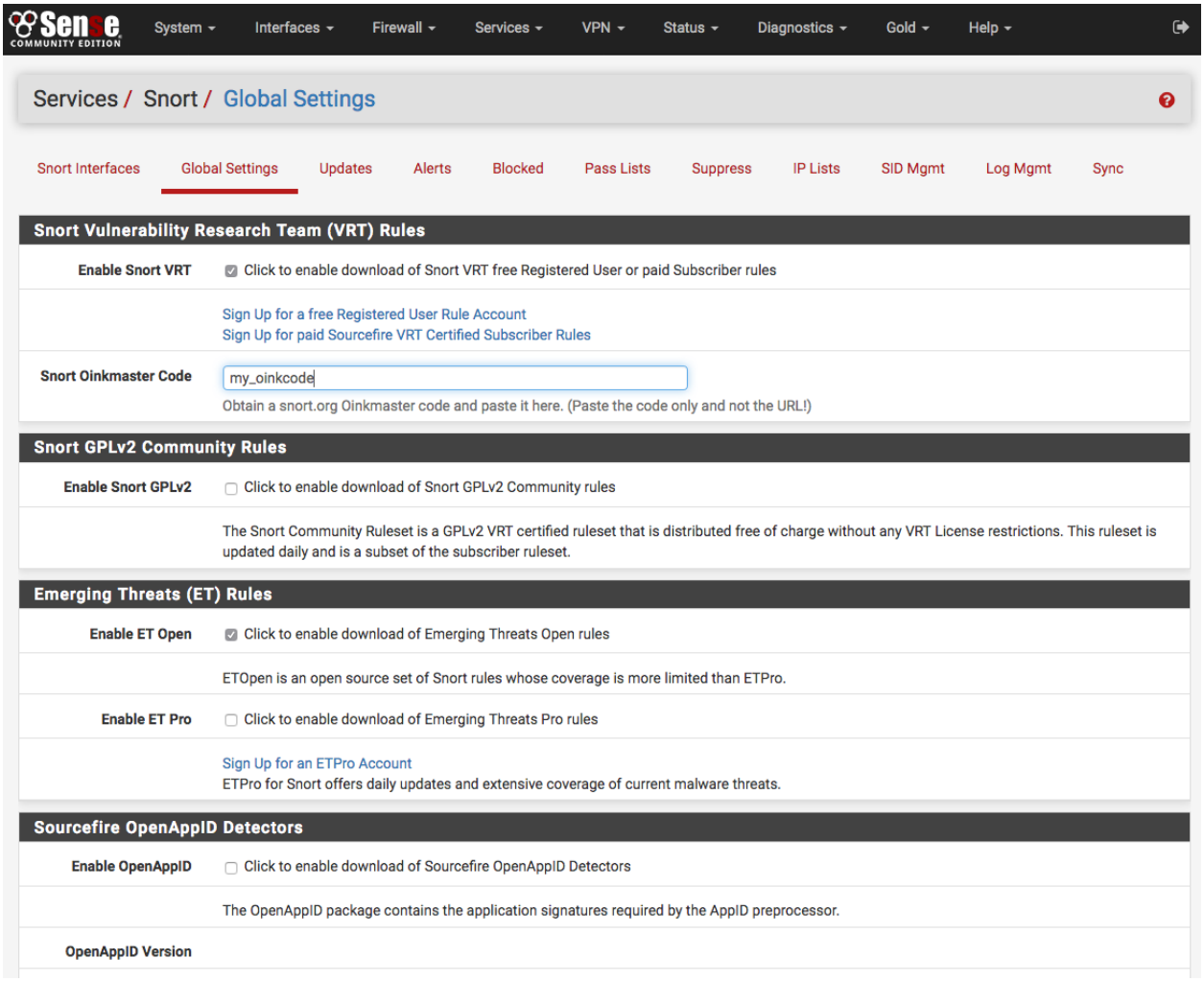
To launch the Snort configuration application, navigate to **Services > Snort** from the menu in the pfSense webGUI.



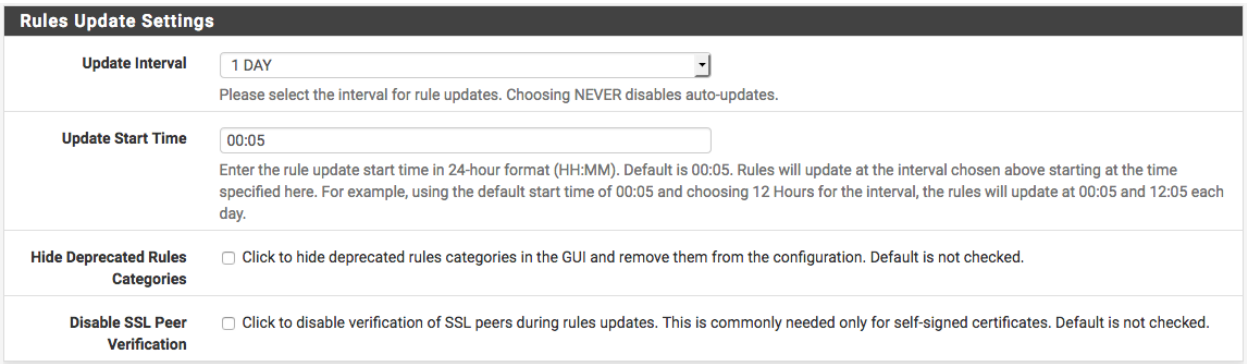
Setting up Snort package for the first time

Click the **Global Settings** tab and enable the rule set downloads to use. If either the Snort VRT or the Emerging Threats Pro rules are checked, a text box will be displayed to enter the unique subscriber code obtained with the subscription or registration.

More than one rule set may be enabled for download, but note the following caveats. If a paid subscription is available for the Snort VRT rules, then all of the Snort GPLv2 Community rules are automatically included within the file downloaded with the Snort VRT rules; therefore, do not enable the GPLv2 Community rules if a paid-subscriber account is used for the Snort VRT rules. All of the Emerging Threats Open rules are included within the paid subscription for the Emerging Threats Pro rules. If the Emerging Threats Pro rules are enabled, the Emerging Threats Open rules are automatically disabled.



Once the desired rule sets are enabled, next set the interval for Snort to check for updates to the enabled rule packages. Use the **Update Interval** drop-down selector to choose a rule update interval. In most cases every *12* hours is a good choice. The update start time may be customized if desired. Enter the time as hours and minutes in 24-hour time format. The default start time is *3* minutes past midnight local time. So with a *12*-hour update interval selected, Snort will check the Snort VRT or Emerging Threats web sites at 3 minutes past midnight and 3 minutes past noon each day for any posted rule package updates.

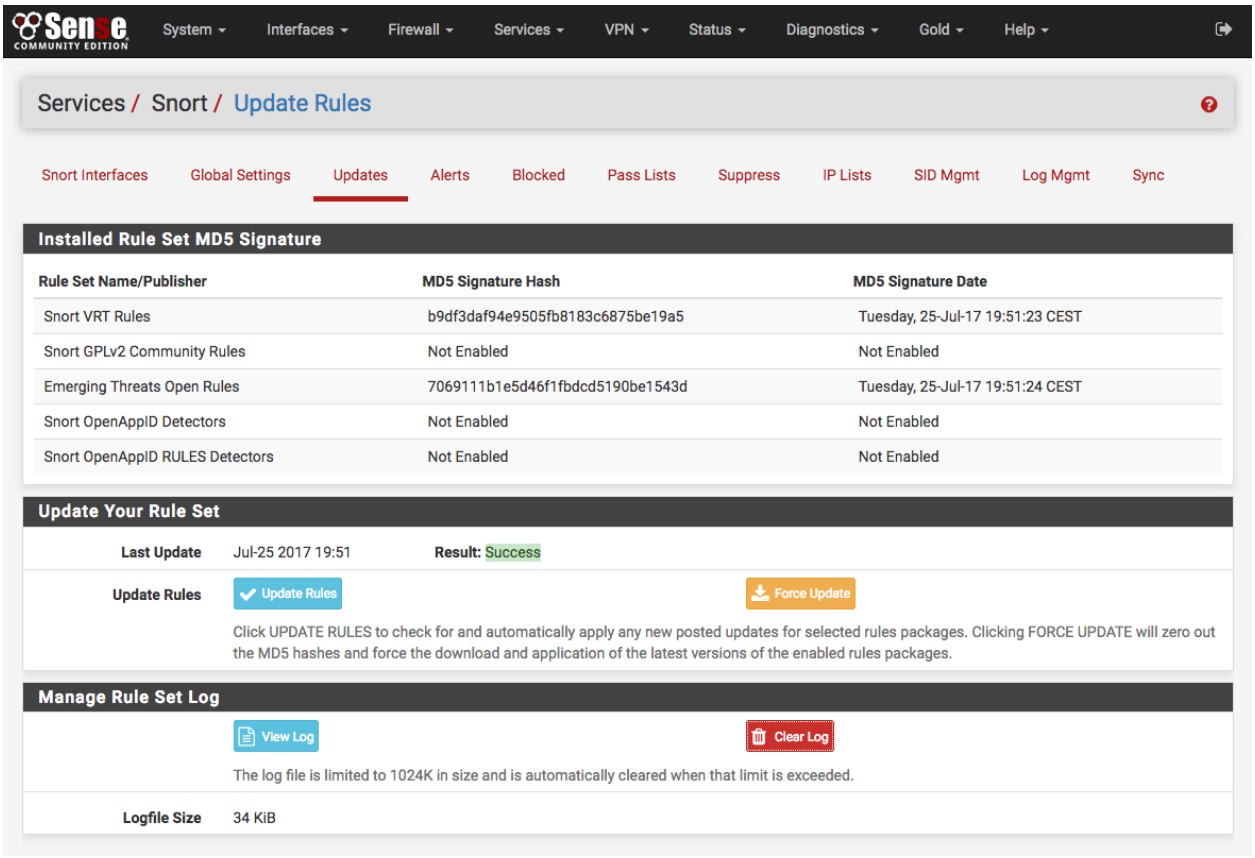


Update the rules

The **Updates** tab is used to check the status of downloaded rules packages and to download new updates. The table shows the available rule packages and their current status (not enabled, not downloaded, or a valid MD5 checksum and date).

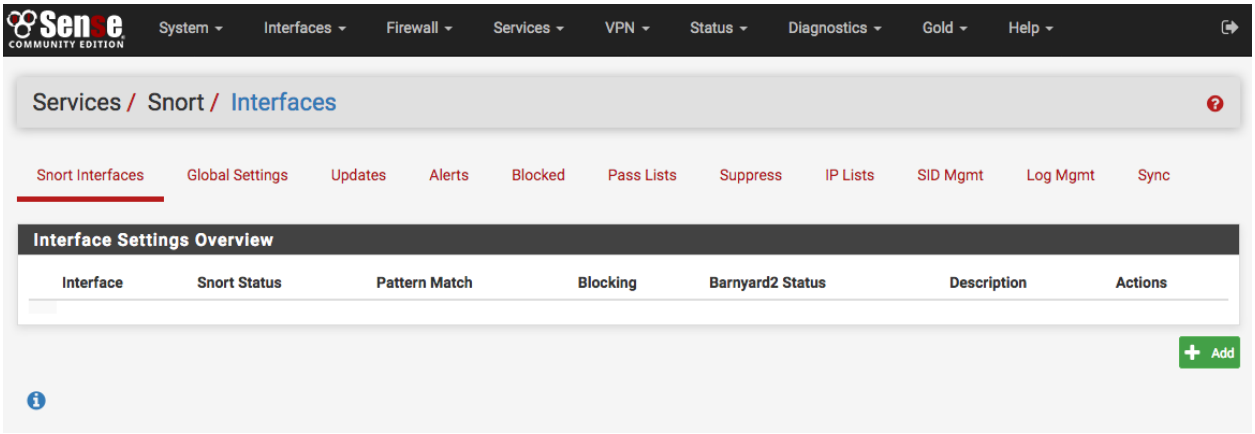
Click on the **Update Rules** button to download the latest rule package updates. If there is a newer set of packaged rules on the vendor web site, it will be downloaded and installed. The determination is made by comparing the MD5 of the local file with that of the remote file on the vendor web site. If there is a mismatch, a new file is downloaded. The **FORCE** button can be used to force download of the rule packages from the vendor web site no matter how the MD5 hash tests out.

In the screenshot below, the Snort VRT and Emerging Threats Open rule packages have been successfully downloaded. The calculated MD5 hash and the file download date and time are shown. Also note the last update time and result are shown in the center of the page.

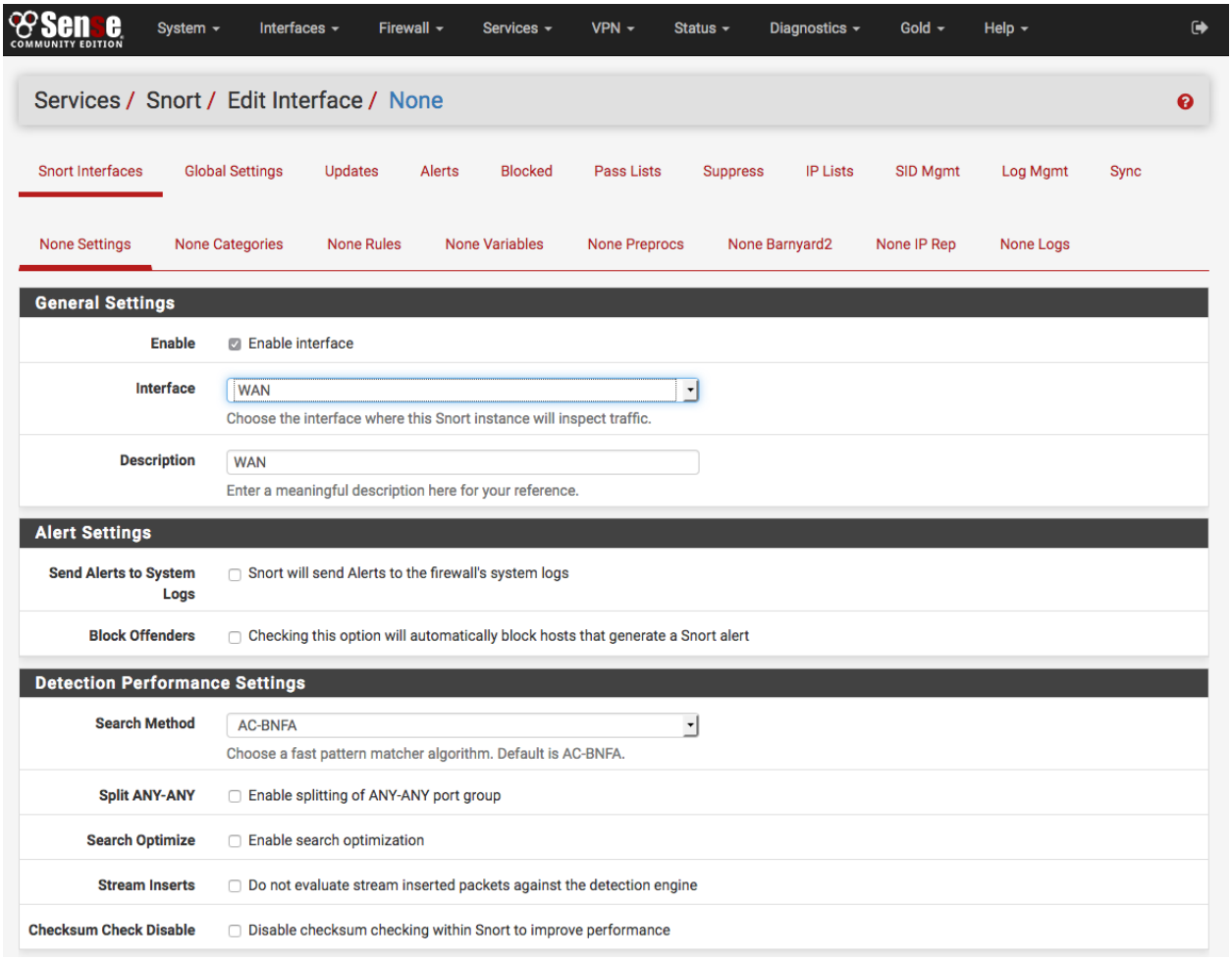


Add Snort to an interface

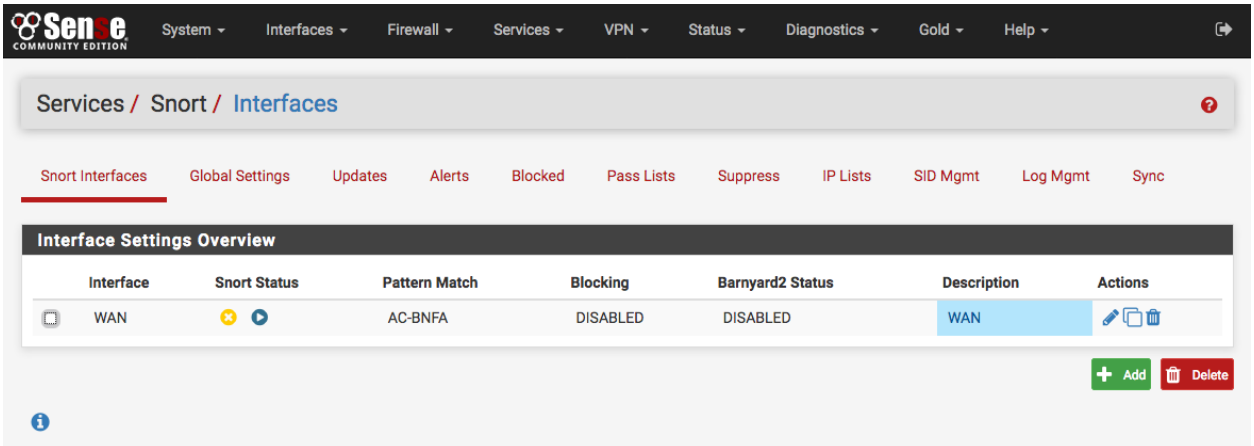
Click the **Snort Interfaces** tab and then the  icon to add a new Snort interface.



A new Interface Settings tab will open with the next available interface automatically selected. The interface selection may be changed using the **Interface** drop-down if desired. A descriptive name may also be provided for the interface. Other interface parameters may also be set on this page. Be sure to click the **SAVE** button down at the bottom of the page when finished.



After saving, the browser will be returned to the **Snort Interfaces** tab. Note the warning icons in the image below showing no rules have been selected for the new Snort interface. Those rules will be configured next. Click the  icon (shown highlighted with a red box in the image below) to edit the new Snort interface again.



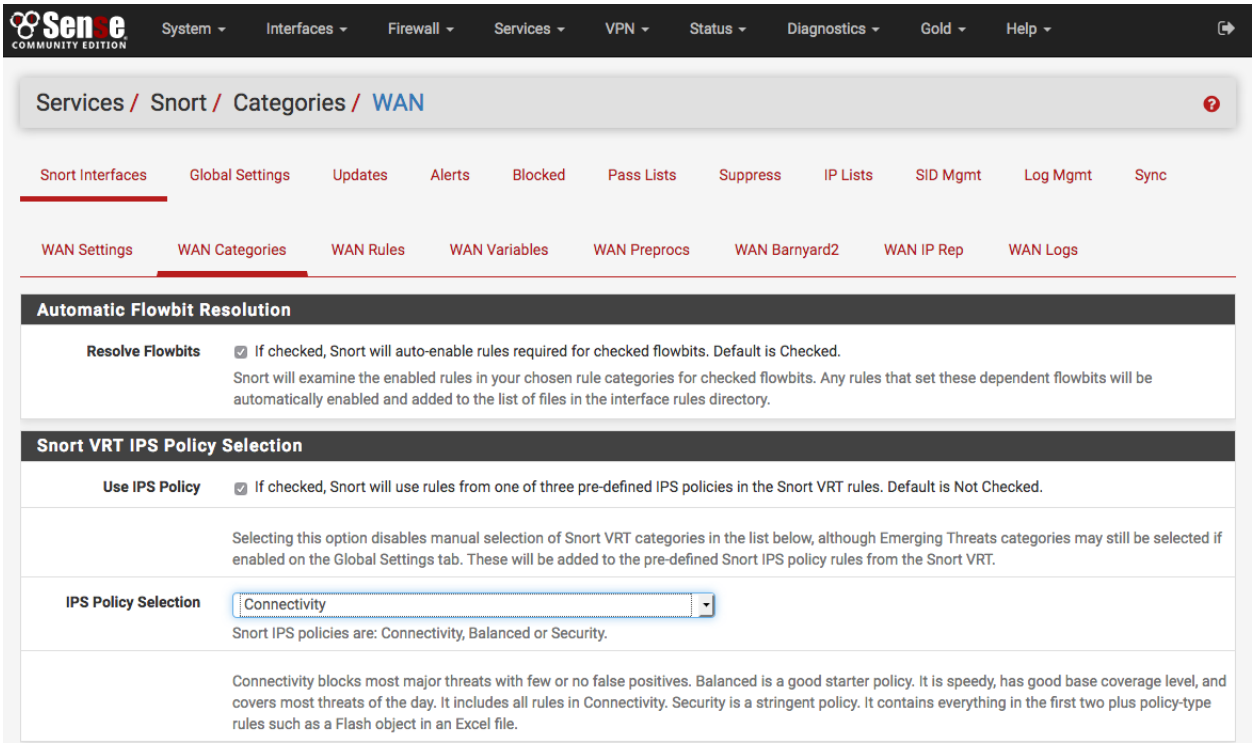
Select which types of rules will protect the network

Click the **Categories** tab for the new interface.

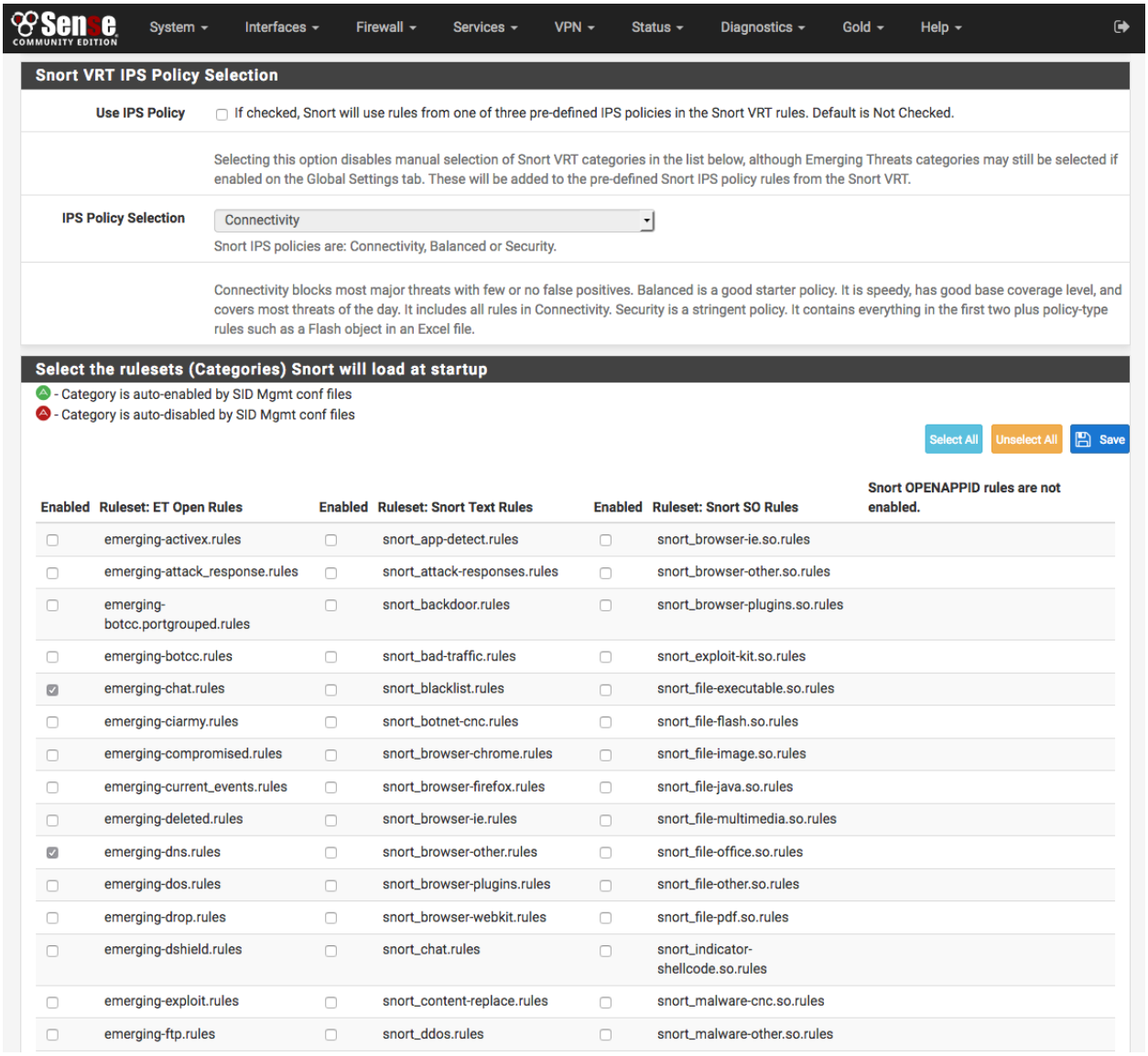
If a Snort VRT Oinkmaster code was obtained (either free registered user or the paid subscription), enabled the Snort VRT rules, and entered the Oinkmaster code on the Global Settings tab then the option of choosing from among three pre-configured IPS policies is available. These greatly simplify the process of choosing enforcing rules for Snort to use when inspecting traffic. The IPS policies are only available when the Snort VRT rules are enabled.

The three Snort VRT IPS Policies are: (1) Connectivity, (2) Balanced and (3) Security. These are listed in order of increasing security. However, resist the temptation to immediately jump to the most secure *Security* policy if Snort is unfamiliar. False positives can frequently occur with the more secure policies, and careful tuning by an experienced administrator may be required.

Tip: If Snort is unfamiliar, then using the less restrictive *Connectivity* policy in non-blocking mode (the default setting) is recommended as a starting point to identify and whitelist false positives. Once experience with Snort has been gained in this network environment, blocking mode may be enabled (via the **Block Offenders** option in the **Snort Interface Settings** tab) and a more restrictive IPS policy may be chosen.



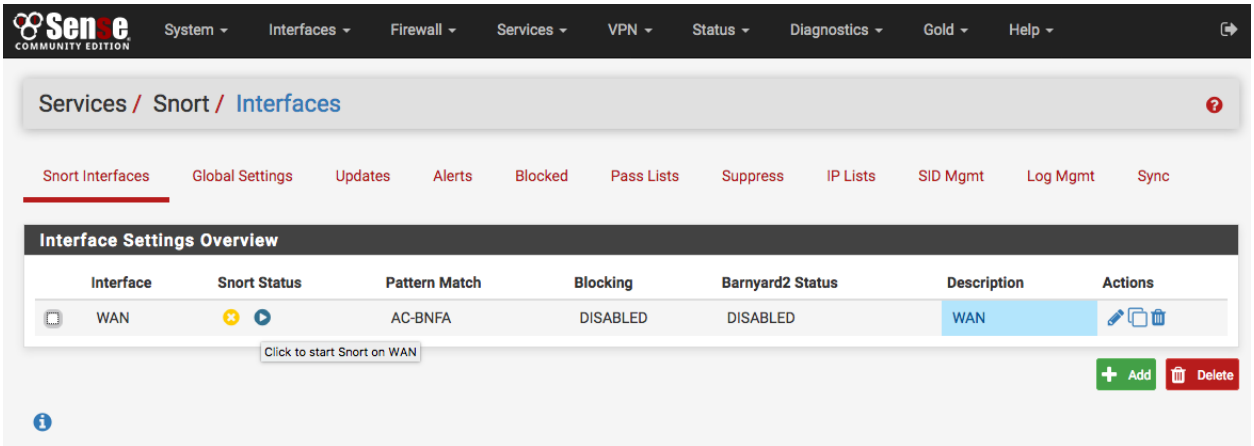
If the Snort VRT rules were not enabled, or if any of the other rule packages are to be used, then make the rule category selections by checking the checkboxes beside the rule categories to use.



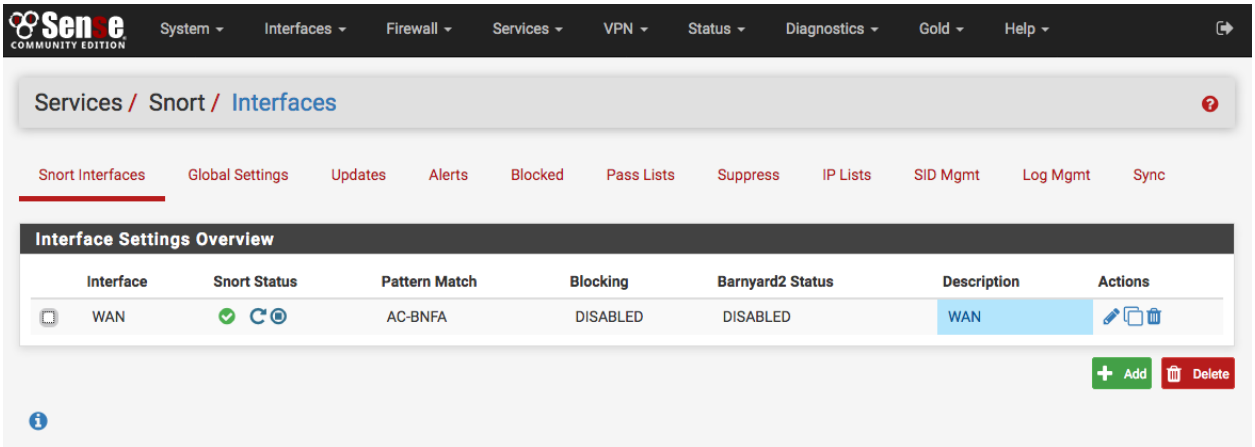
Be sure to click **SAVE** when finished to save the selection and build the rules file for Snort to use.

Starting Snort on an interface

Click the **Snort Interfaces** tab to display the configured Snort interfaces. Click the  icon (shown highlighted with a red box in the image below) to start Snort on an interface.



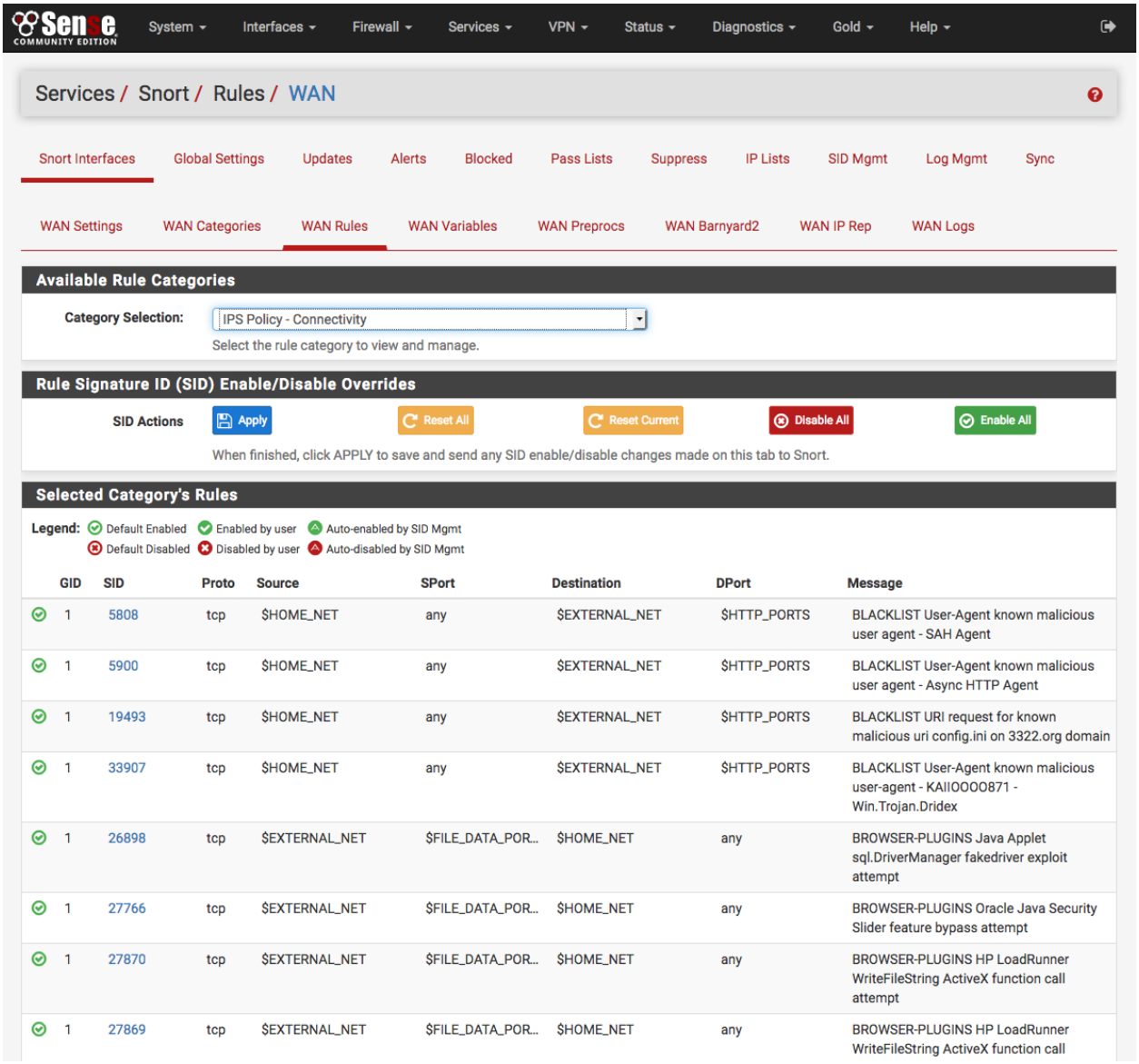
It will take several seconds for Snort to start. Once it has started, the icon will change to as shown below. To stop a running Snort instance on an interface, click the icon.

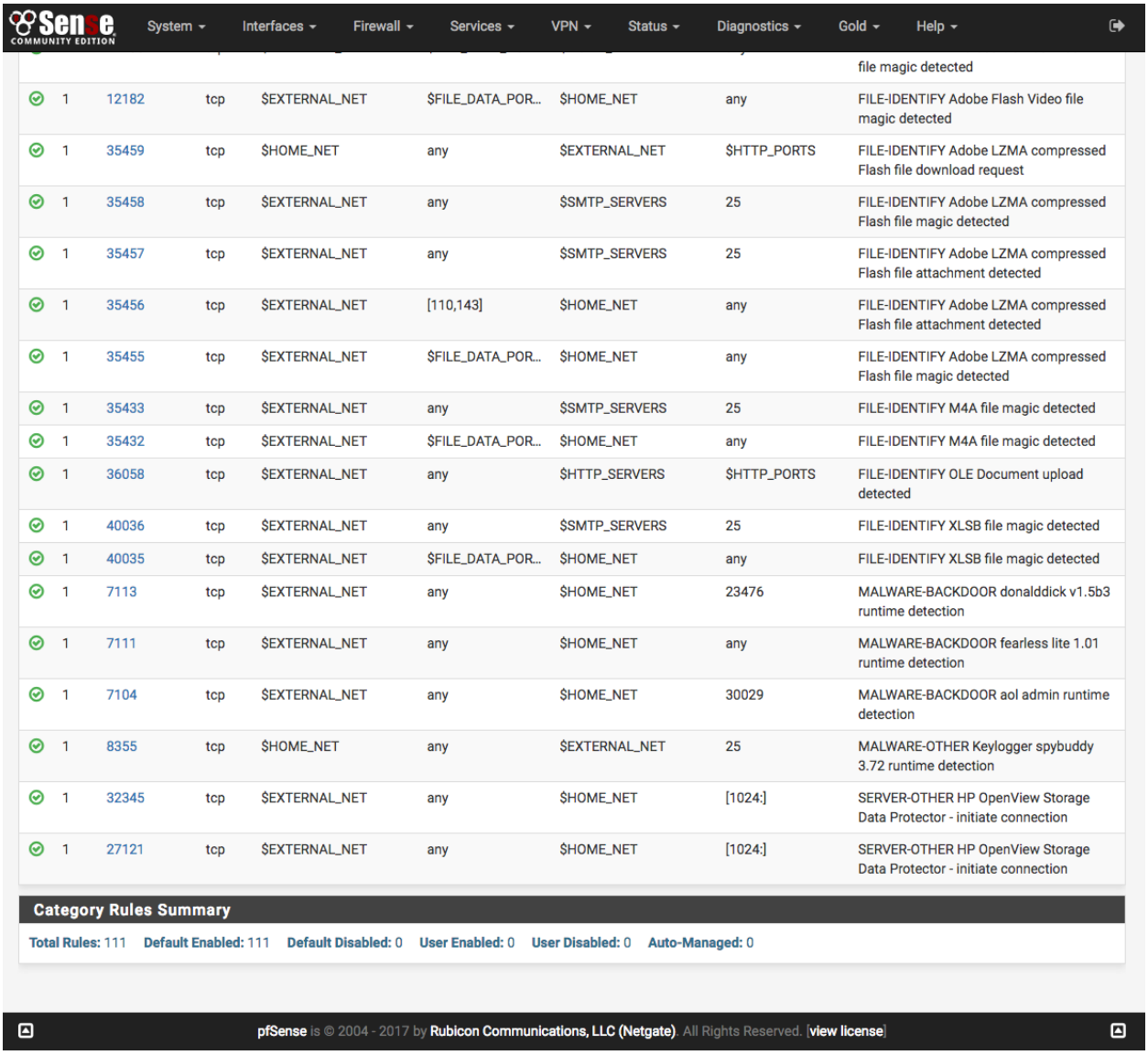


Select which types of signatures will protect the network

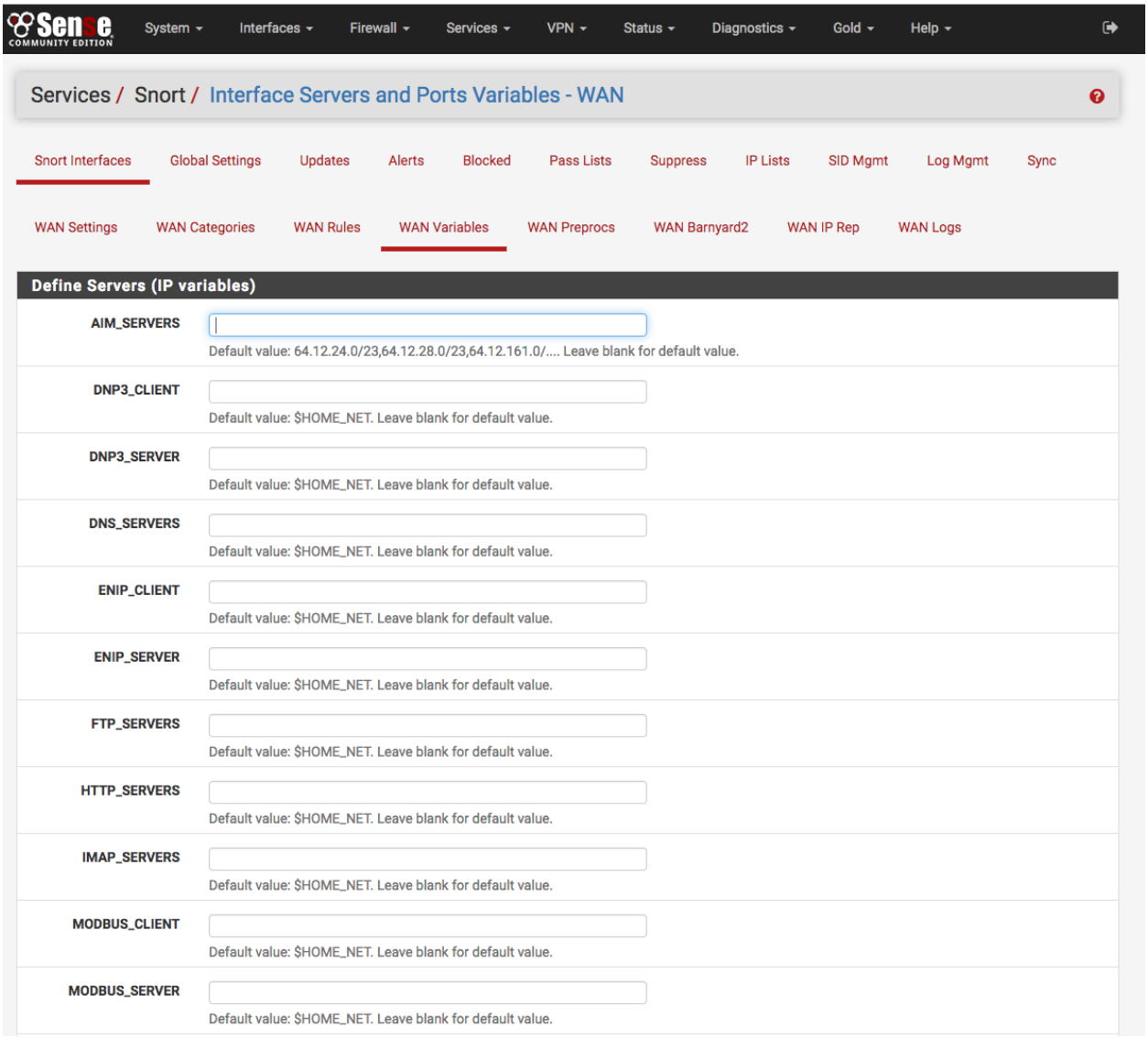
Click the **Rules** tab for the interface to configure individual rules in the enabled categories. Generally this page is only used to disable particular rules that may be generating too many false positives in a particular network environment. Be sure they are in fact truly false positives before taking the step of disabling a Snort rule!

Select a rules category from the **Category** drop-down to view all the assigned rules. Click the or icon at the far-left of a row to toggle the rule's state from enabled to disabled, or click or to toggle from disabled to enabled. The icon will change to indicate the state of the rule. At the top of the rule list is a legend showing the icons used to indicate the current state of a rule.


pfSense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Gold
Help
Services / Snort / Rules / WAN
Snort Interfaces
Global Settings
Updates
Alerts
Blocked
Pass Lists
Suppress
IP Lists
SID Mgmt
Log Mgmt
Sync
WAN Settings
WAN Categories
WAN Rules
WAN Variables
WAN Preprocs
WAN Barnyard2
WAN IP Rep
WAN Logs
Available Rule Categories
Category Selection:
IPS Policy - Connectivity
Select the rule category to view and manage.
Rule Signature ID (SID) Enable/Disable Overrides
SID Actions
Apply
Reset All
Reset Current
Disable All
Enable All
When finished, click APPLY to save and send any SID enable/disable changes made on this tab to Snort.
Selected Category's Rules
Legend:
Default Enabled
Enabled by user
Auto-enabled by SID Mgmt
Default Disabled
Disabled by user
Auto-disabled by SID Mgmt
GID SID Proto Source SPort Destination DPort Message
1 5808 tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS BLACKLIST User-Agent known malicious user agent - SAH Agent
1 5900 tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS BLACKLIST User-Agent known malicious user agent - Async HTTP Agent
1 19493 tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS BLACKLIST URI request for known malicious uri config.ini on 3322.org domain
1 33907 tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS BLACKLIST User-Agent known malicious user-agent - KAIIOOOO871 - Win.Trojan.Dridex
1 26898 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any BROWSER-PLUGINS Java Applet sql.DriverManager fakedriver exploit attempt
1 27766 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any BROWSER-PLUGINS Oracle Java Security Slider feature bypass attempt
1 27870 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any BROWSER-PLUGINS HP LoadRunner WriteFileString ActiveX function call attempt
1 27869 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any BROWSER-PLUGINS HP LoadRunner WriteFileString ActiveX function call

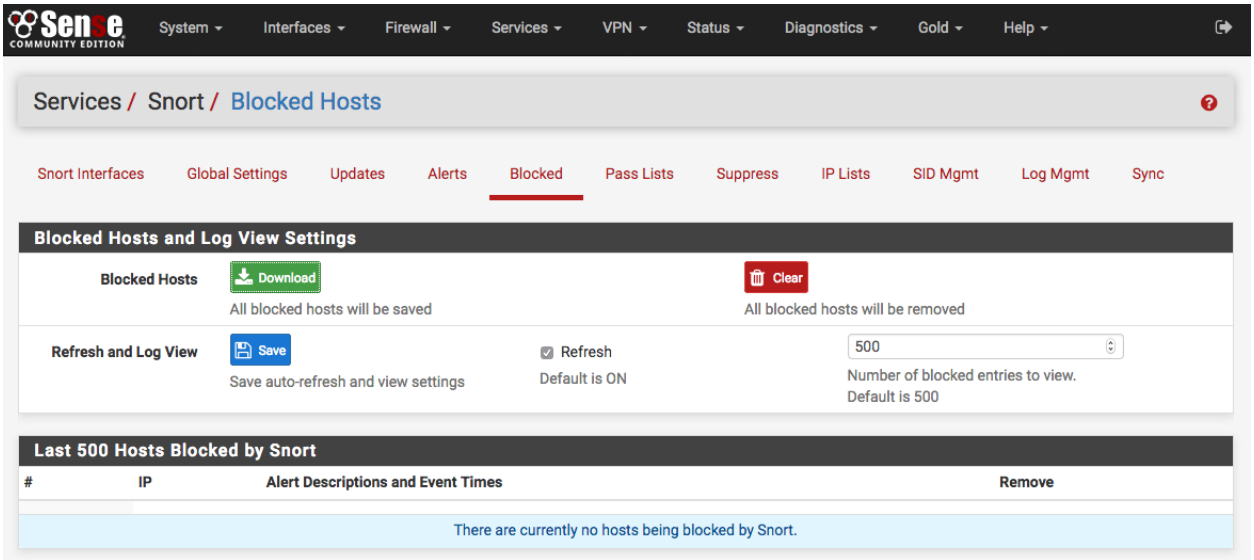

pfSense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Gold
Help
file magic detected
1 12182 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any FILE-IDENTIFY Adobe Flash Video file magic detected
1 35459 tcp $HOME_NET any $EXTERNAL_NET $HTTP_PORTS FILE-IDENTIFY Adobe LZMA compressed Flash file download request
1 35458 tcp $EXTERNAL_NET any $SMTP_SERVERS 25 FILE-IDENTIFY Adobe LZMA compressed Flash file magic detected
1 35457 tcp $EXTERNAL_NET any $SMTP_SERVERS 25 FILE-IDENTIFY Adobe LZMA compressed Flash file attachment detected
1 35456 tcp $EXTERNAL_NET [110,143] $HOME_NET any FILE-IDENTIFY Adobe LZMA compressed Flash file attachment detected
1 35455 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any FILE-IDENTIFY Adobe LZMA compressed Flash file magic detected
1 35433 tcp $EXTERNAL_NET any $SMTP_SERVERS 25 FILE-IDENTIFY M4A file magic detected
1 35432 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any FILE-IDENTIFY M4A file magic detected
1 36058 tcp $EXTERNAL_NET any $HTTP_SERVERS $HTTP_PORTS FILE-IDENTIFY OLE Document upload detected
1 40036 tcp $EXTERNAL_NET any $SMTP_SERVERS 25 FILE-IDENTIFY XLSB file magic detected
1 40035 tcp $EXTERNAL_NET $FILE_DATA_POR... $HOME_NET any FILE-IDENTIFY XLSB file magic detected
1 7113 tcp $EXTERNAL_NET any $HOME_NET 23476 MALWARE-BACKDOOR donalddick v1.5b3 runtime detection
1 7111 tcp $EXTERNAL_NET any $HOME_NET any MALWARE-BACKDOOR fearless lite 1.01 runtime detection
1 7104 tcp $EXTERNAL_NET any $HOME_NET 30029 MALWARE-BACKDOOR aol admin runtime detection
1 8355 tcp $HOME_NET any $EXTERNAL_NET 25 MALWARE-OTHER Keylogger spybuddy 3.72 runtime detection
1 32345 tcp $EXTERNAL_NET any $HOME_NET [1024:] SERVER-OTHER HP OpenView Storage Data Protector - initiate connection
1 27121 tcp $EXTERNAL_NET any $HOME_NET [1024:] SERVER-OTHER HP OpenView Storage Data Protector - initiate connection
Category Rules Summary
Total Rules: 111 Default Enabled: 111 Default Disabled: 0 User Enabled: 0 User Disabled: 0 Auto-Managed: 0
pfSense is © 2004 - 2017 by Rubicon Communications, LLC (Netgate). All Rights Reserved. [view license]

Define servers to protect and improve performance



Managing blocked hosts

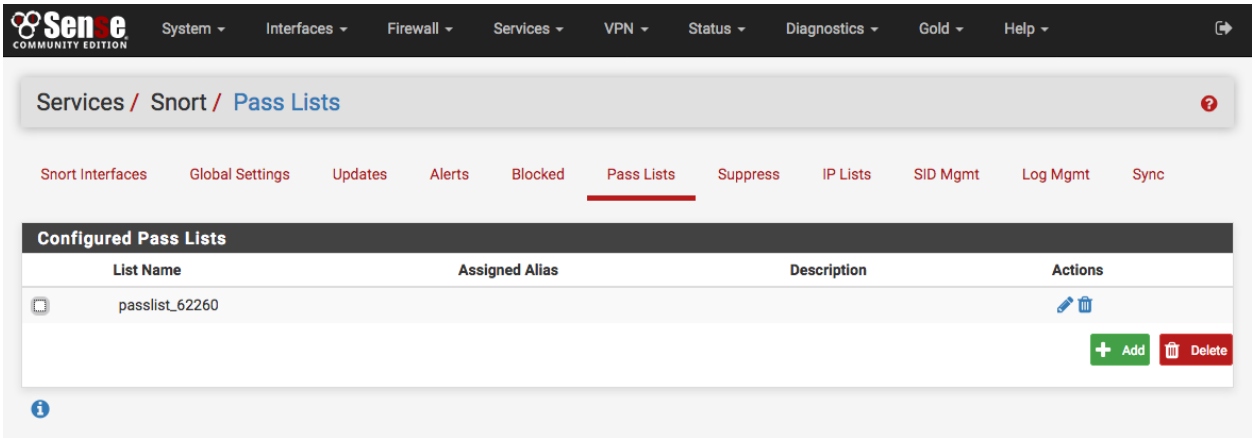
The **Blocked** tab shows what hosts are currently being blocked by Snort (when the block offenders option is selected on the **Interface Settings** tab). Blocked hosts can be automatically cleared by Snort at one of several pre-defined intervals. The blocking options for an interface are configured on the Snort **Interface Settings** tab for the interface.



Managing Pass lists

Pass Lists are lists of IP addresses that Snort should never block. These may be created and managed on the **Pass Lists** tab. When an IP address is listed on a Pass List, Snort will never insert a block on that address even when malicious traffic is detected.

To create a new Pass List, click  . To edit an existing Pass List, click the  . To delete a Pass List, click  . Note that a Pass List may not be deleted if it is currently assigned to one or more Snort interfaces.

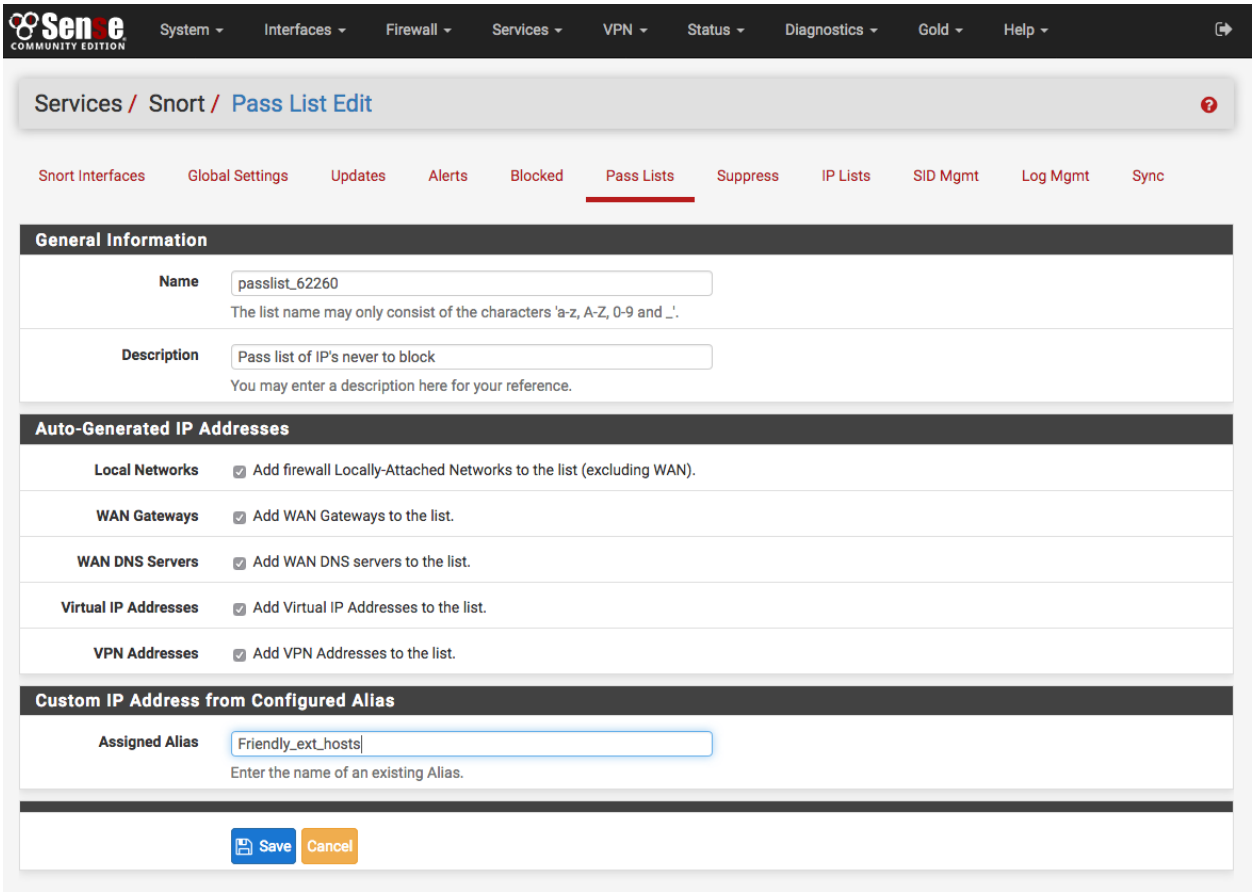


A default Pass List is automatically generated by Snort for every interface, and this default list is used when no other list is specified. Pass Lists are assigned to an interface on the **Interface Settings** tab.

Customized Pass List may be created and assigned to an interface. This might be done when trusted external hosts exist that are not located on networks directly connected to the firewall. To add external hosts in this manner, first create an Alias under **Firewall > Aliases** and then assign that alias to the **Assigned Aliases** field. In the example shown below, the alias "*Friendly_ext_hosts*" has been assigned. This alias would contain the IP addresses of the trusted external hosts.

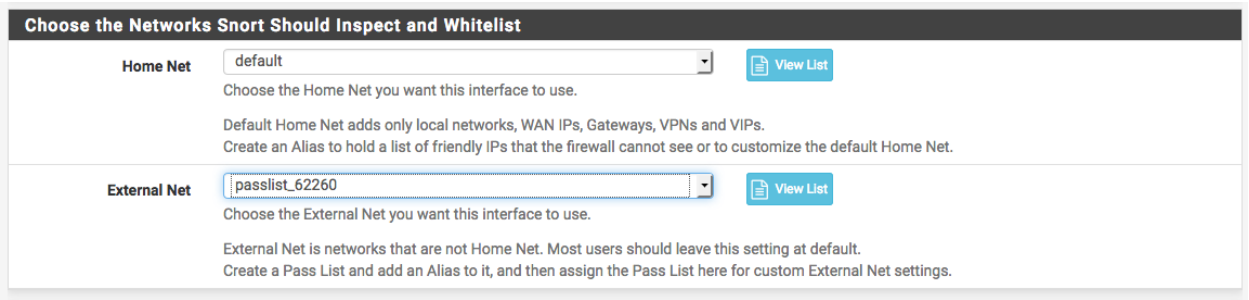
When creating a custom Pass List, leave all the auto-generated IP addresses checked in the **Add auto-generated IP**

addresses section. Not selecting the checkboxes in this section can lead to blocking of critical addresses including the firewall interfaces themselves. This could result in being locked out of the firewall over the network! Only uncheck boxes in this section when absolutely necessary.



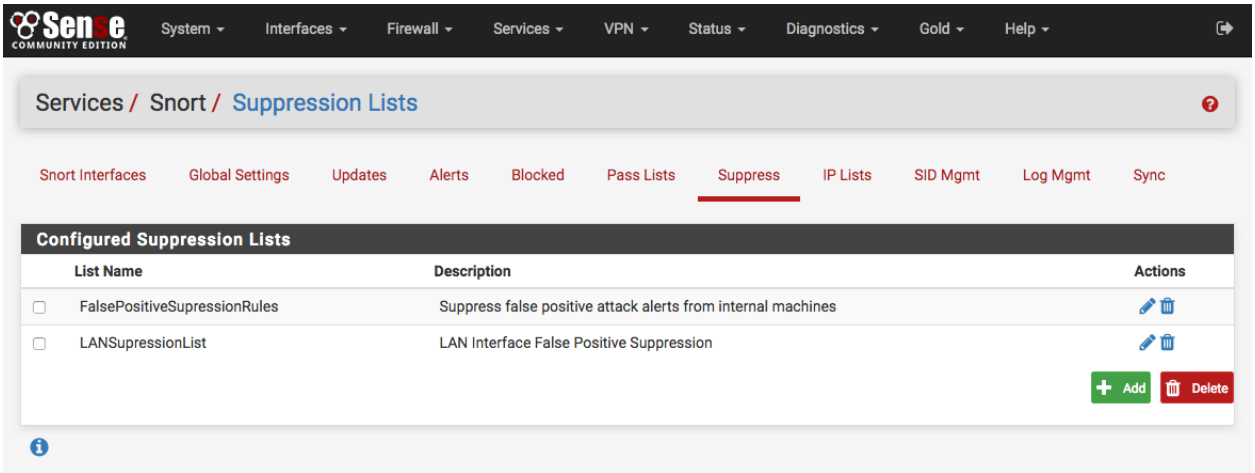
Click the **ALIASES** button to open a window showing previously defined aliases for selection. Remember to click **SAVE** to save changes.

Note: Remember that simply creating a Pass List is only the first step! It must be selected by going to the **Interface Settings** tab for the Snort interface and assigning the newly created Pass List as shown below. After assigning and saving the new Pass List, restart Snort on the affected interface to pick up the change.

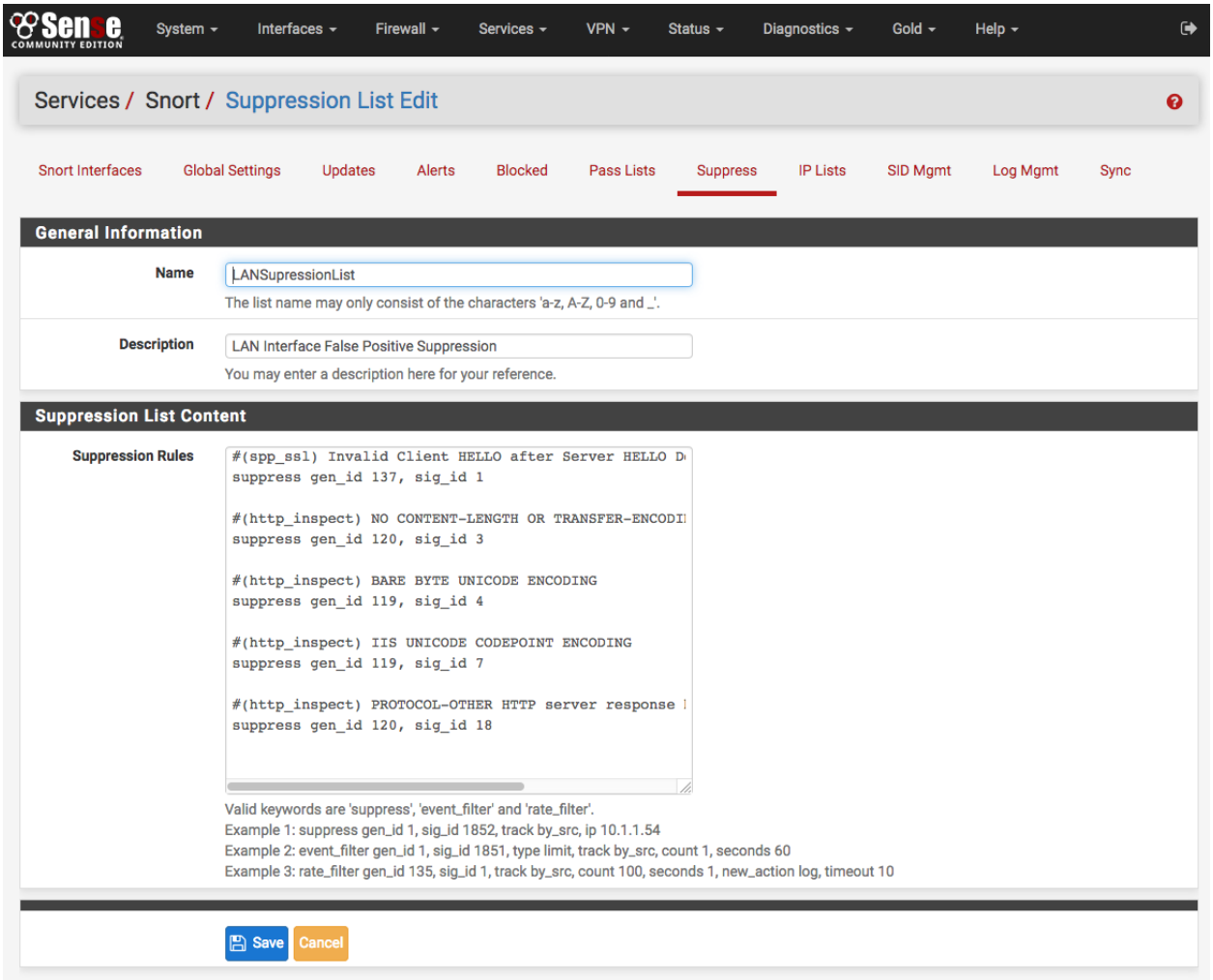


Alert Thresholding and Suppression

Suppression Lists allow control over the alerts generated by Snort rules. When an alert is suppressed, then Snort no longer logs an alert entry (or blocks the IP address if block offenders is enabled) when a particular rule fires. Snort still inspects all network traffic against the rule, but even when traffic matches the rule signature, no alert will be generated. This is different from disabling a rule. When a rule is disabled, Snort no longer tries to match it to any network traffic. Suppressing a rule might be done in lieu of disabling the rule when alerts should only be stopped based on either the source or destination IP. For example, to suppress the alert when traffic from a particular trusted IP address is the source. If any other IP is the source or destination of the traffic, the rule would still fire. To eliminate all alerts from the rule, then it is more efficient to simply disable the rule rather than to suppress it. Disabling the rule will remove it from Snort's list of match rules and therefore makes for less work Snort has to do.



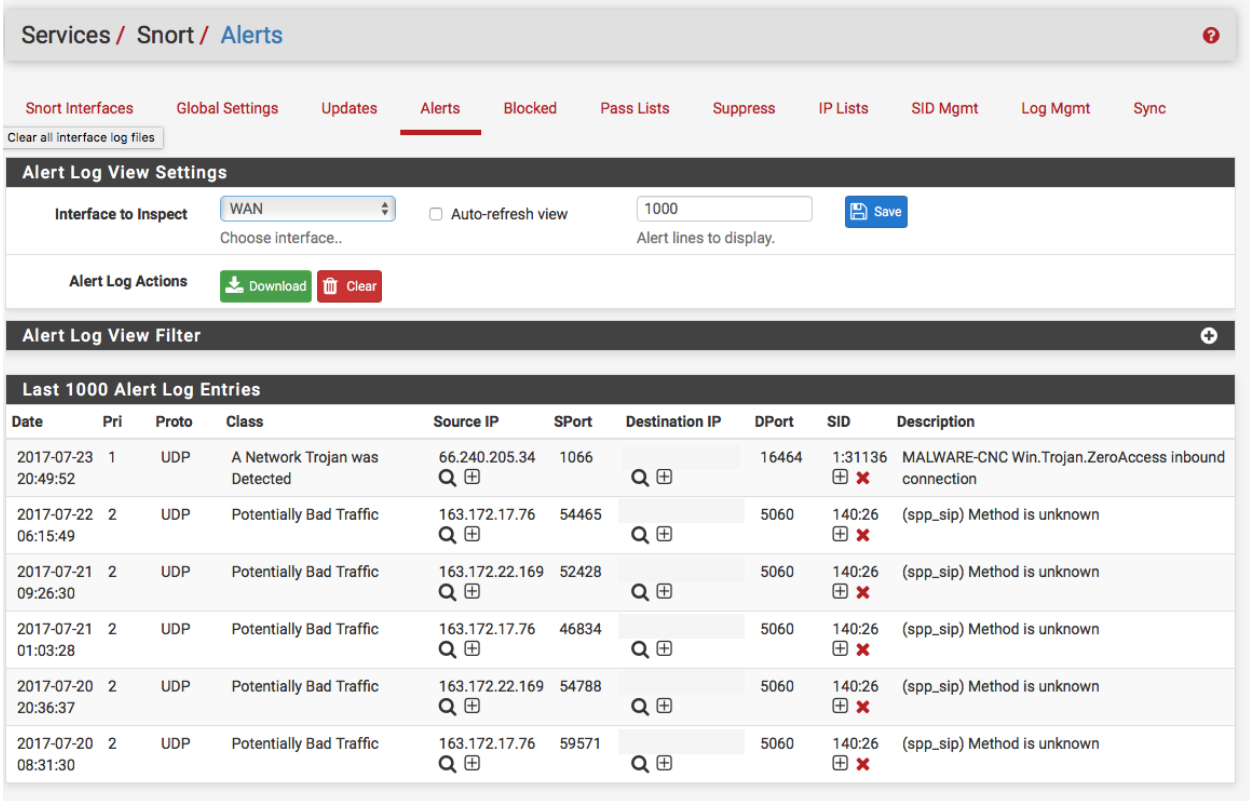
On the Suppress List Edit page, a new suppress list entry may be manually added or edited. It is usually easier and faster to add suppress list entries by clicking  shown with the alert entries on the **Alerts** tab. Remember to click the **SAVE** button to save changes when manually editing Suppress List entries.



Getting to know the alerts

The **Alerts** tab is where alerts generated by Snort are viewed. If Snort is running on more than one interface, choose the interface whose alerts should be viewed in the drop-down selector.

Use the **DOWNLOAD** button to download a gzip tar file containing all of the logged alerts to a local machine. The **CLEAR** button is used to erase the current alerts log. Destination IP's have been redacted from the screenshot.



Alert Details

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066		16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465		5060	140:26	(spp_sip) Method is unknown

The **Date** column shows the date and time the alert was generated. The remaining columns show data from the rule that generated the alert.

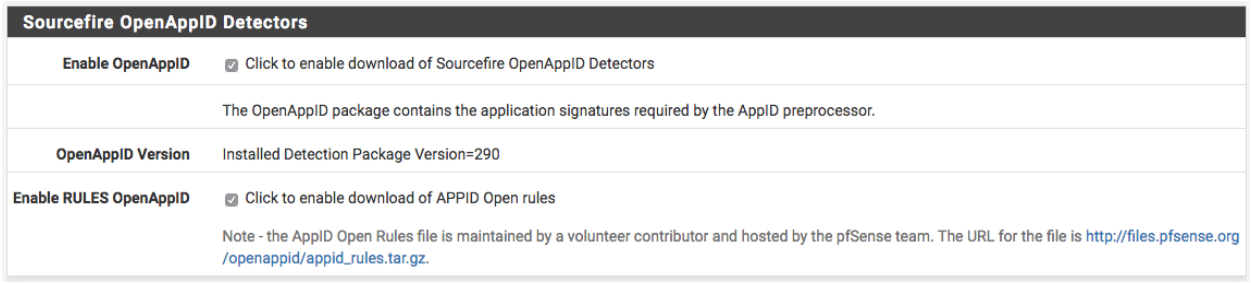
In the **Source**, **Destination** columns are  icons for performing reverse DNS lookups on the IP addresses as well as a  icon used to add an automatic *Suppress List* entry for the alert using the IP address and SID (signature ID). This will prevent future alerts from being generated by the rule for that specific IP address only. If either of the Source or Destination addresses are currently being blocked by Snort, then a  icon will also be shown. Clicking that icon will remove the block for the IP address.

The SID column contains two icons. The  icon will automatically add that SID to the *Suppress List* for the interface and suppress future alerts from the signature for all IP addresses. The  icon in the SID column will disable the rule and remove it from the enforcing rule set. When a rule is manually disabled, the icon in the SID column changes to  .

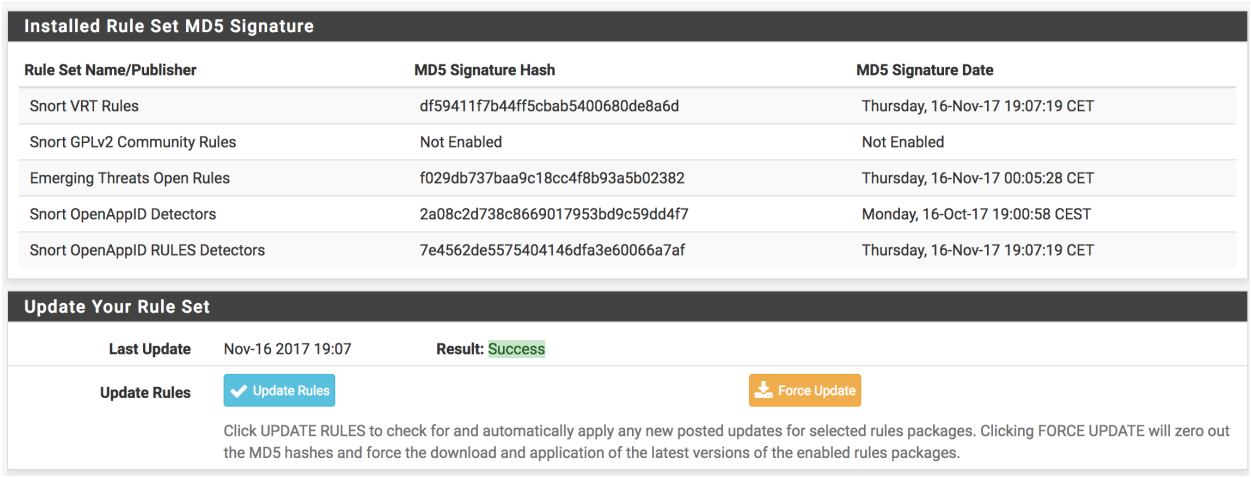
Application ID detection with OpenApp ID

OpenAppID is an application-layer network security plugin for the open source intrusion detection system Snort. Learn more about it here.

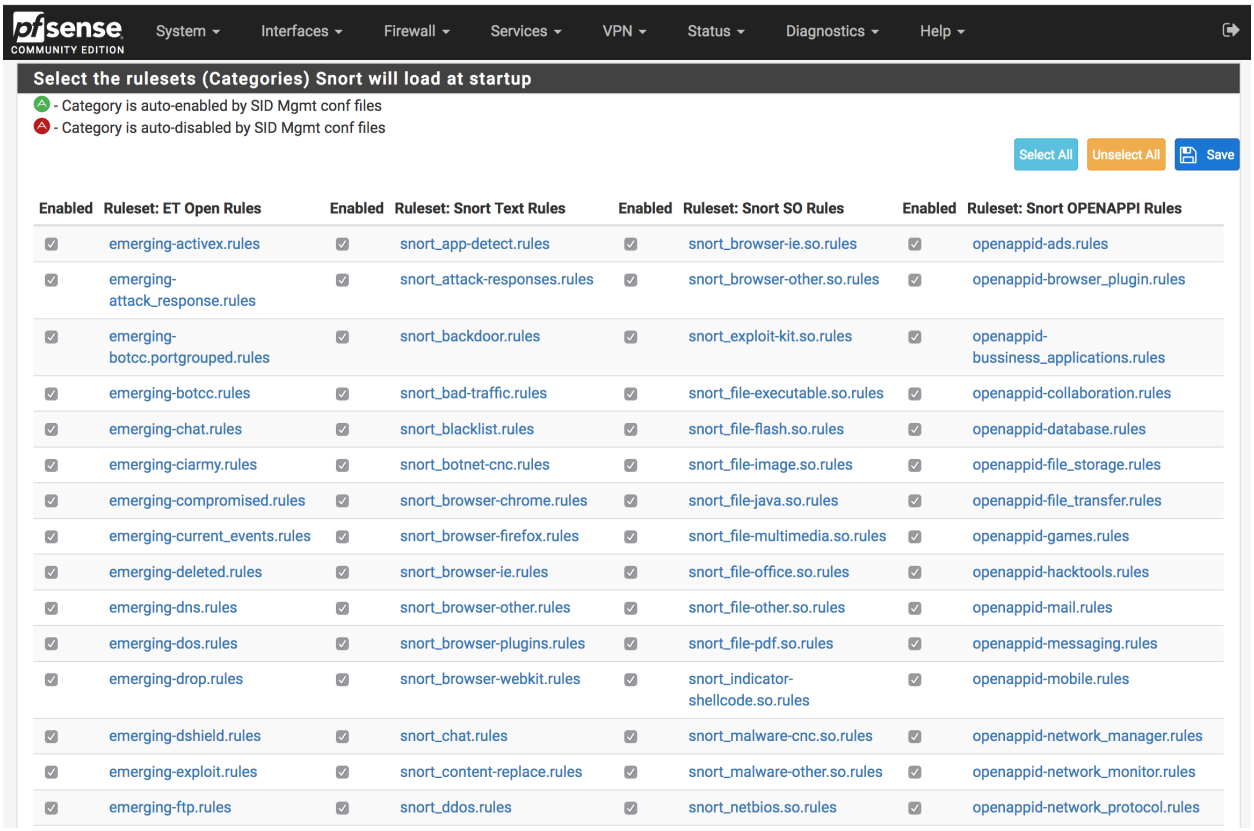
Enabling OpenAppID and its rules is done from Snort **Global Settings**. Select both checkboxes to enable detectors and rules download. Save the page.



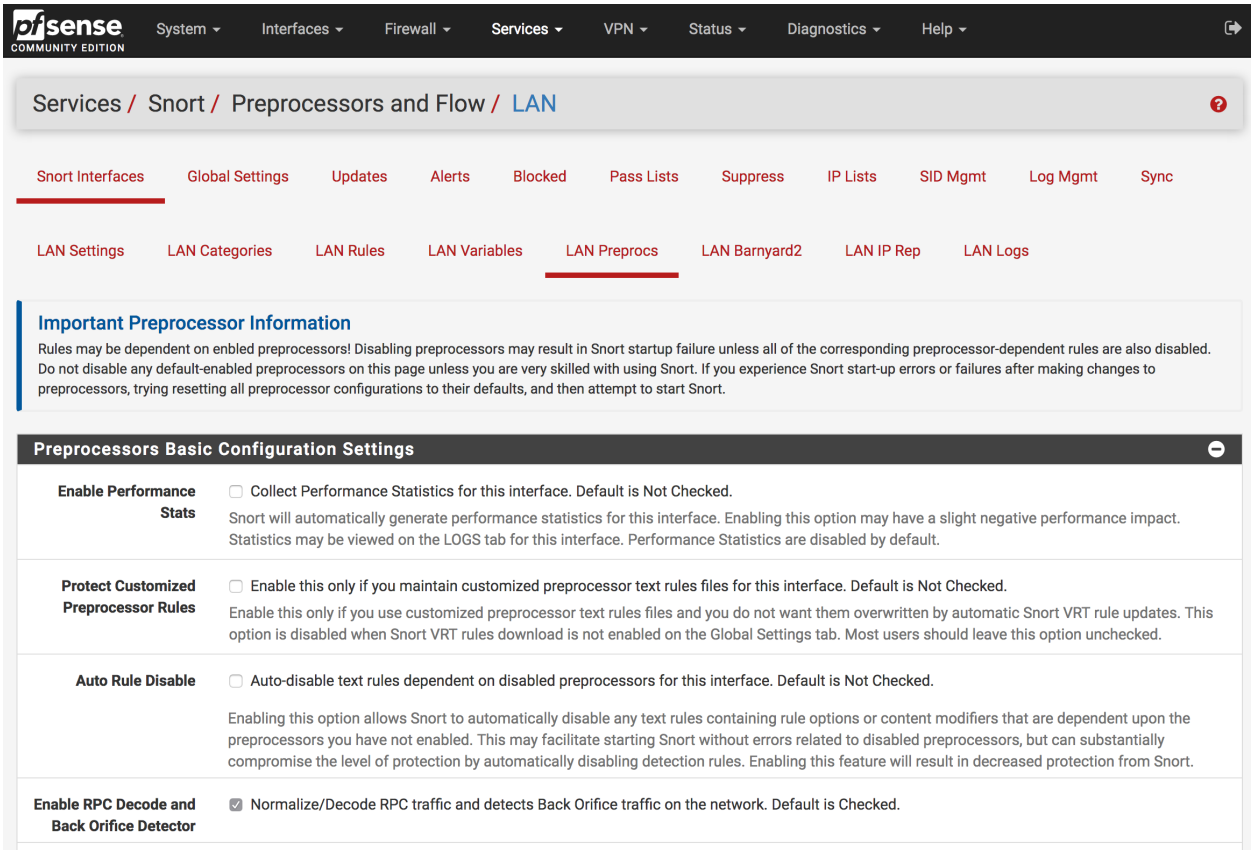
After enabling the detectors and rules go to Snort Updates tab and click on **Update Rules**. Wait for all the rules to update. Once done, the page will show OpenAppID detectors and rules have been updated.



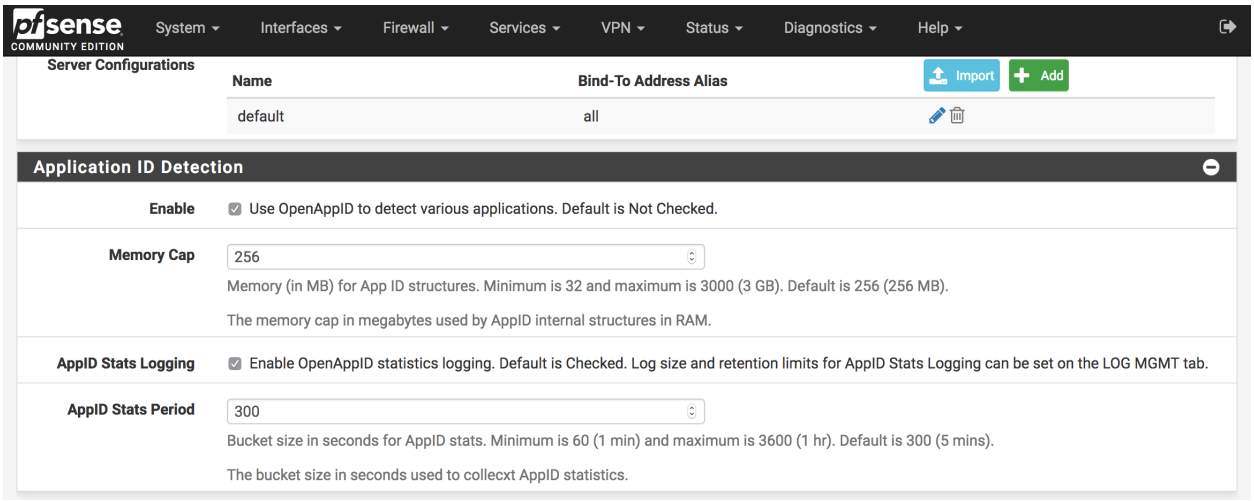
The following steps assume the firewall already has a Snort interface for LAN. Edit the LAN interface and navigate to LAN categories tab. When there, make sure the **Snort OPENAPPID Rules** from the right column are all selected and click **Save**.



Lastly, while still editing Snort interface, navigate to **LAN Preprocessor** tab.

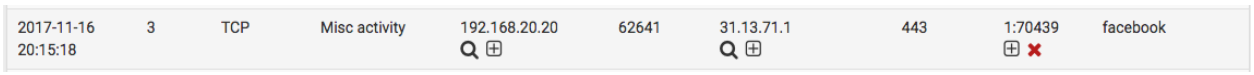


Scroll down to **Application ID Detection** section and select both **Enable** and **AppID Stats Logging** checkboxes. Save the page the OpenApp ID will be activated on the Snort interface.



Viewing detected applications can be done from **Alerts** tab. The following screenshots are examples of identified services and applications:

Facebook



Netflix

2017-11-16 20:09:28	3	TCP	Misc activity	192.168.20.9	59412	45.57.45.159	80	1:70542	netflix

Reddit

2017-11-16 20:14:28	3	TCP	Misc activity	192.168.20.20	62623	151.101.129.140	443	1:70588	reddit

Amazon Web Services

2017-11-16 20:11:14	3	TCP	Misc activity	192.168.20.9	34759	52.94.212.133	443	1:70012	Amazon Webservices

iCloud

2017-11-16 20:13:20	3	TCP	Misc activity	192.168.20.20	62595	17.248.146.110	443	1:70904	icloud

Twitter

2017-11-16 20:21:03	3	TCP	Misc activity	192.168.20.20	62654	104.244.46.71	443	1:70656	twitter

Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

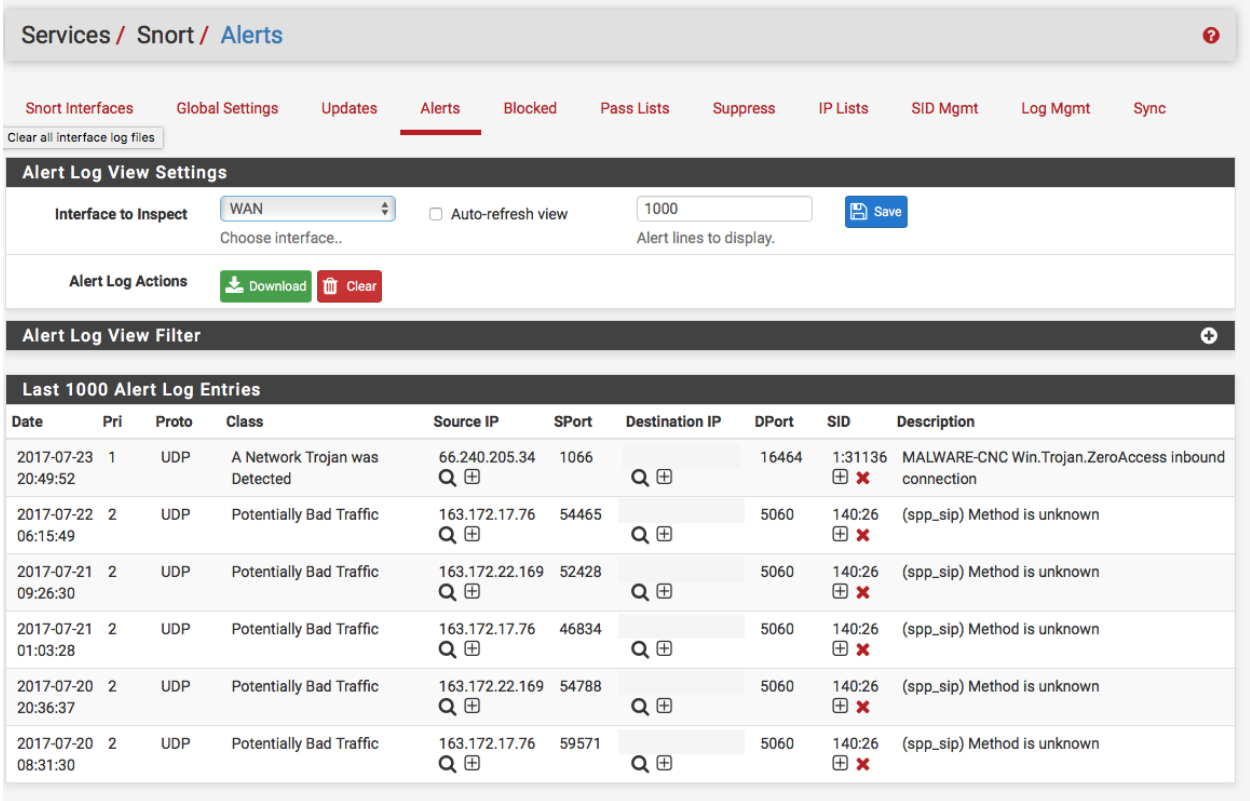
Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

Snort Alerts

The **Alerts** tab is where alerts generated by Snort may be viewed. If Snort is running on more than one interface, choose the interface to view alerts for in the drop-down selector.

Use the **DOWNLOAD** button to download a gzip tar file containing all of the logged alerts to a local machine. The **CLEAR** button is used to erase the current alerts log.



Alert Details

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066		16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465		5060	140:26	(spp_sip) Method is unknown

The **Date** column shows the date and time the alert was generated. The remaining columns show data from the rule that generated the alert.

In the **Source**, **Destination** columns are  icons for performing reverse DNS lookups on the IP addresses as well as a  icon used to add an automatic *Suppress List* entry for the alert using the IP address and SID (signature ID). This will prevent future alerts from being generated by the rule for that specific IP address only. If either of the Source or Destination addresses are currently being blocked by Snort, then a  icon will also be shown. Clicking that icon will remove the block for the IP address.

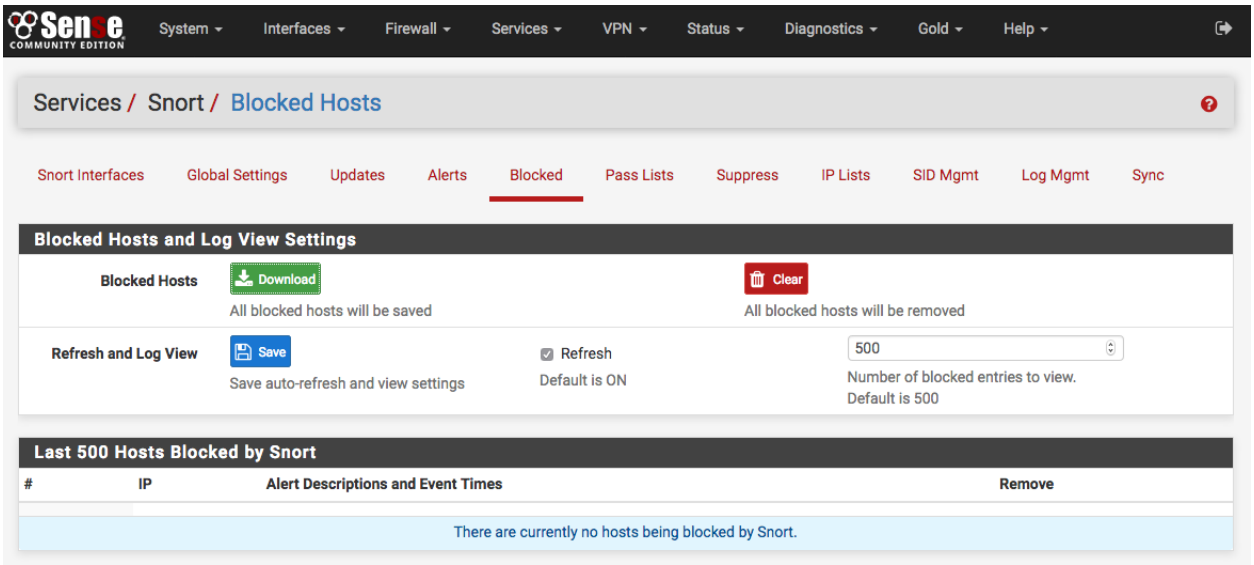
The SID column contains two icons. The  icon will automatically add that SID to the *Suppress List* for the interface and suppress future alerts from the signature for all IP addresses. The  icon in the SID column will disable the rule and remove it from the enforcing rule set. When a rule is manually disabled, the icon in the SID column changes to  .

Snort Blocked Hosts

The **Blocked** tab shows what hosts are currently being blocked by Snort (when the **block offenders** option is selected on the **Interface Settings** tab). Blocked hosts can be automatically cleared by Snort at one of several pre-defined intervals. The blocking options for an interface are configured on the Snort **Interface Settings** tab for the interface.

To manually remove a blocked host, click the  icon in the right-hand column.

The  icon performs a reverse DNS lookup on the blocked host IP address when clicked.



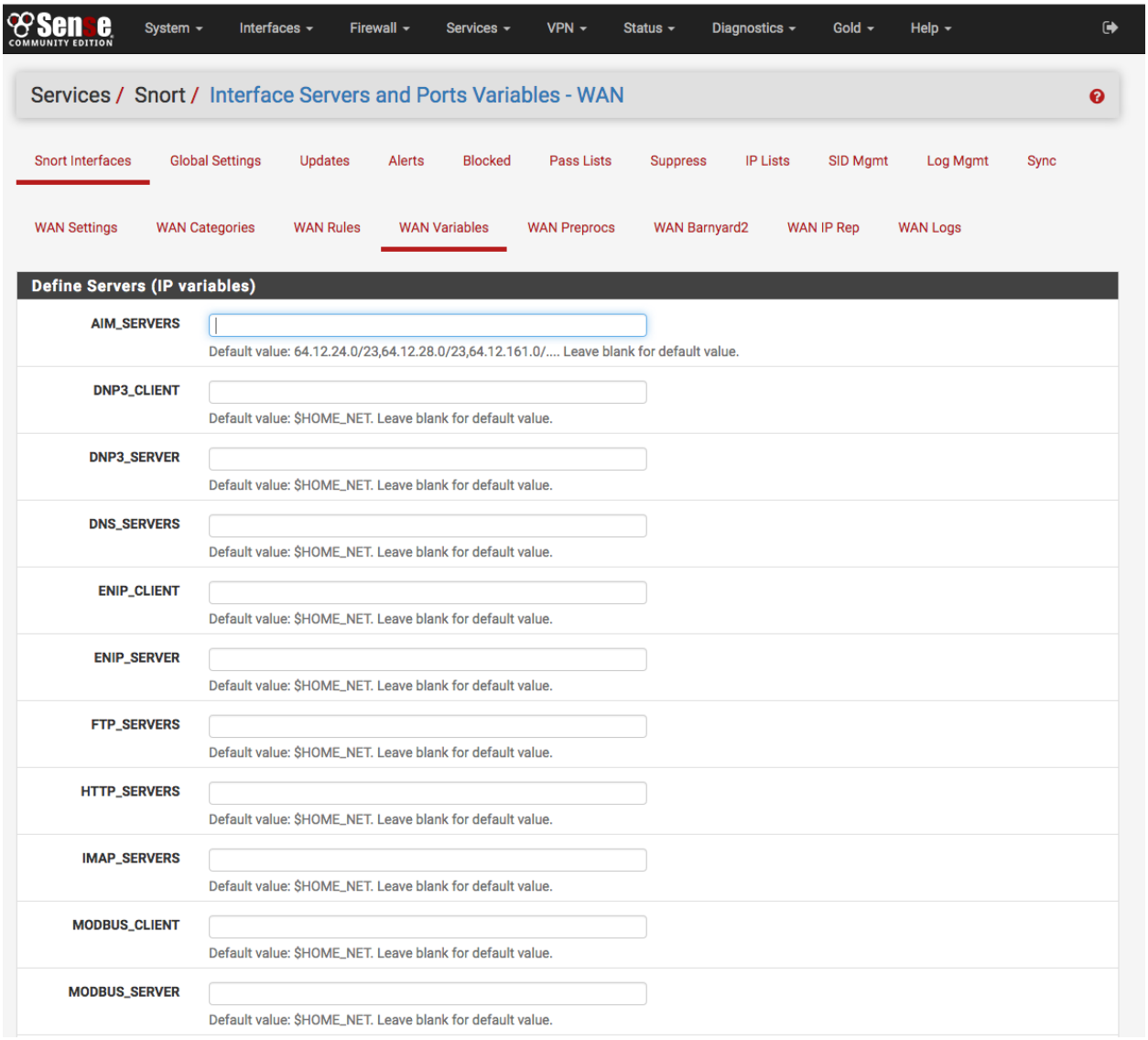
Snort Server Definitions

Define servers to protect and improve performance

The **Variables** tab is where the specific types of hosts on the network are configured. For example, the specific IP addresses or network ranges containing web servers to protect may be defined. This can make Snort more efficient because it won't waste time scanning for web server threats on IP addresses where web servers do not exist. Similarly, Snort performance can be optimized by instructing it which addresses contain other critical servers such as SMTP, POP, DNS, etc.

The exact ports or port ranges used for certain services on the network may also be specified.

Each value entered on this page can only be an existing Alias. Start typing the name of the Alias into a textbox and a drop-down selection of matching entries will appear for selection. Aliases are created under **Firewall > Aliases** from the menu.



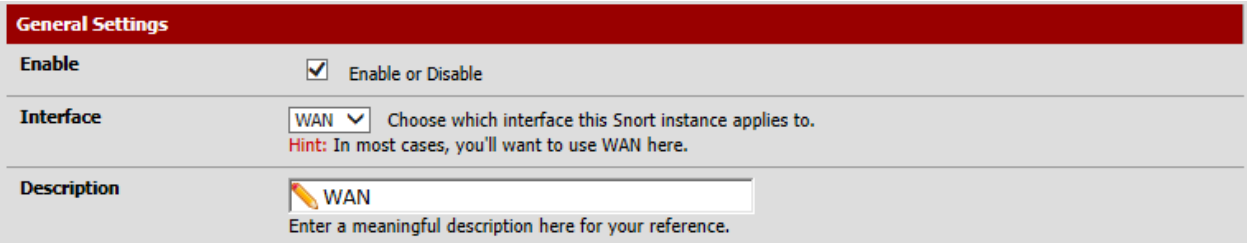
Snort interface Settings

General Settings

Enable: used to enable or disable Snort on the selected interface. Snort is enabled on the interface when this box is checked.

Interface: used to choose which physical firewall interface this Snort instance protects.

Description: used to provide an optional friendly name for the interface.



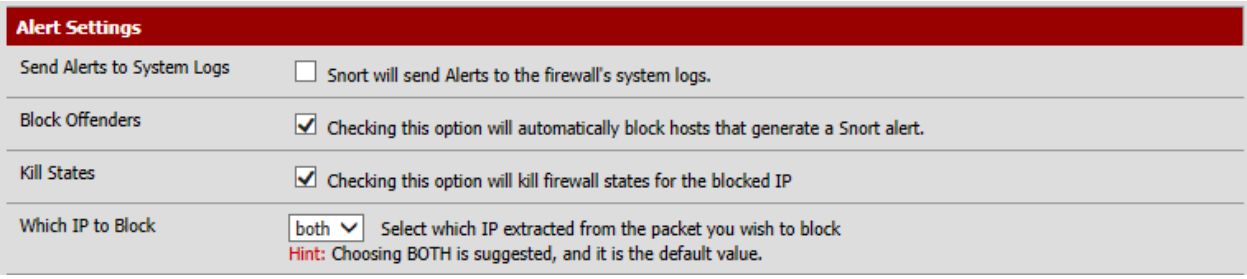
Alert Settings

Send Alerts to System Logs: when checked, all Snort alerts will be copied to the system log on the firewall.

Block Offenders: when checked, Snort will automatically insert a firewall block of the host generating an alert.

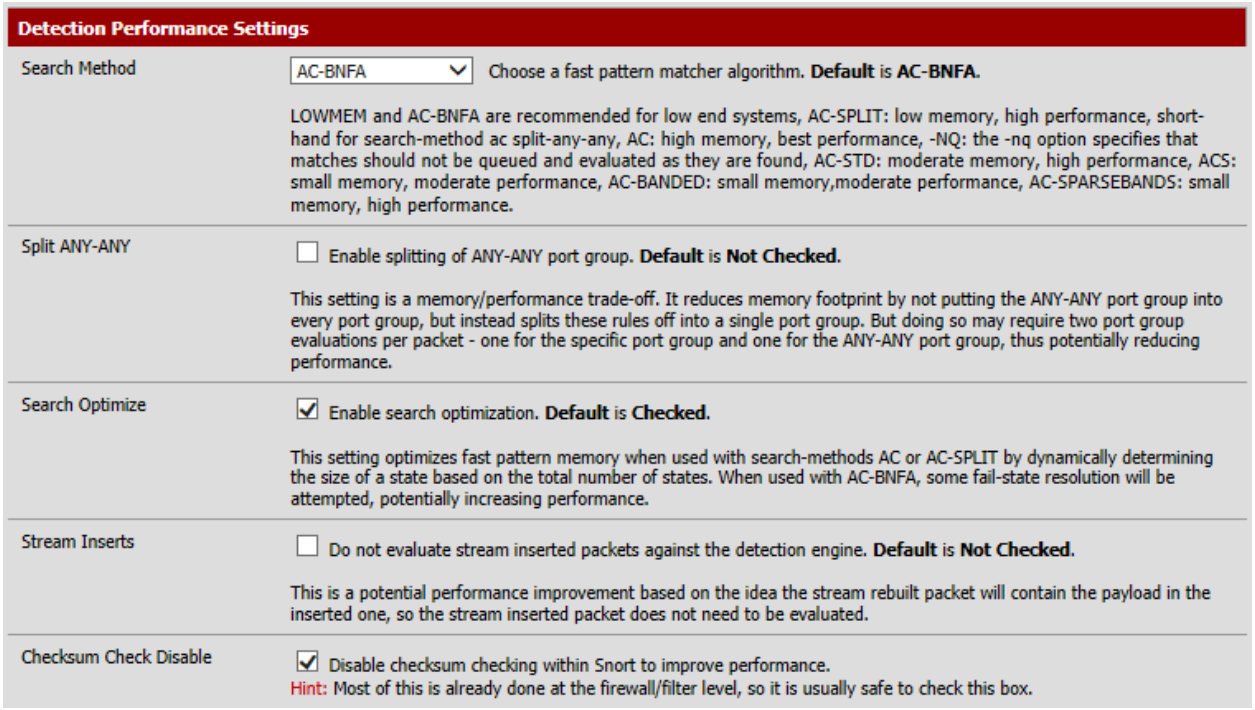
Kill States: when checked, Snort will kill all existing state table entries for the IP address it blocks. This should generally be enabled (box checked).

Which IP to Block: this determines which IP address extracted from the packet that generated an alert will be blocked. The choices are SOURCE, DESTINATION or BOTH. BOTH is the recommended default.



Detection Performance Settings

Search Method: used to select the pattern matcher algorithm used by Snort in the signature detection engine.

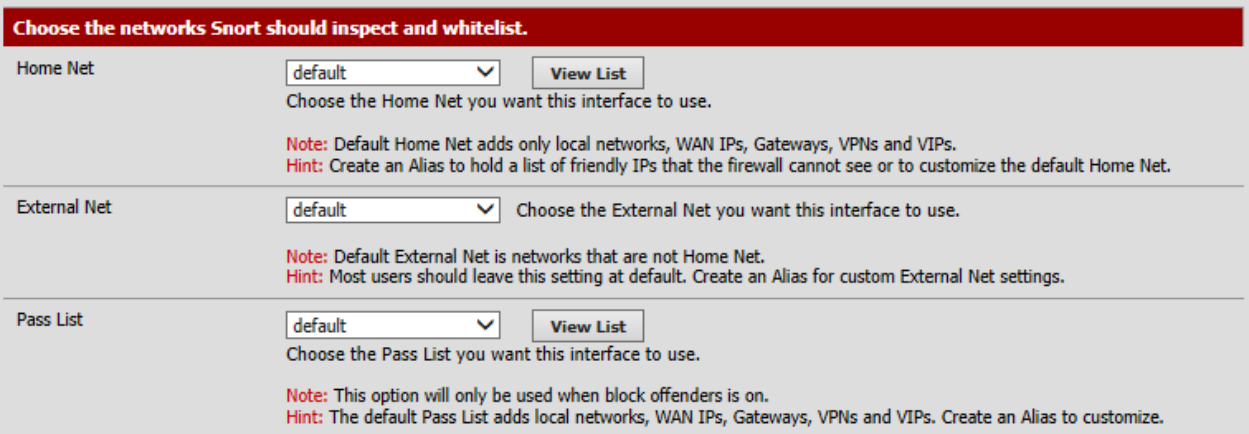


Choose the networks Snort should inspect and whitelist

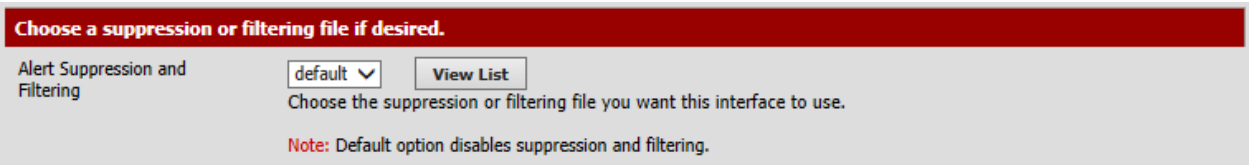
Home Net: selects the network Snort will use as the HOME_NET variable. Default is the recommended choice and contains the firewall WAN IP address and WAN gateway, all networks locally-attached to a firewall interface, the configured DNS servers, VPN addresses and Virtual IP addresses. Additional HOME_NET networks may be created on the IP LISTS tab, and then return to this tab to assign them to the Snort interface by selecting the appropriate list in the drop-down selector. View the contents of the selected list by clicking the **View List** button.

External Net: selects the network will use as the EXTERNAL_NET variable. Default is the recommended choice and contains all networks not included in HOME_NET. Create additional EXTERNAL_NET networks on the IP LISTS tab, and then return to this tab to assign them to the Snort interface by selecting the appropriate list in the drop-down selector.

Pass List: selects the networks and IP addresses that Snort will never block. These represent "trusted hosts". Even if a trusted host generates a Snort alert, it will not be blocked if the IP address is on a Pass List. The default Pass List contains the same addresses as HOME_NET. Create additional pass lists on the IP LISTS tab, and then return to this tab to assign them to the Snort interface by selecting the appropriate list in the drop-down selector. Snort must be restarted on the interface when making changes to the Pass List. View the contents of the selected list by clicking the **View List** button.



Choose a suppression or filtering file if desired

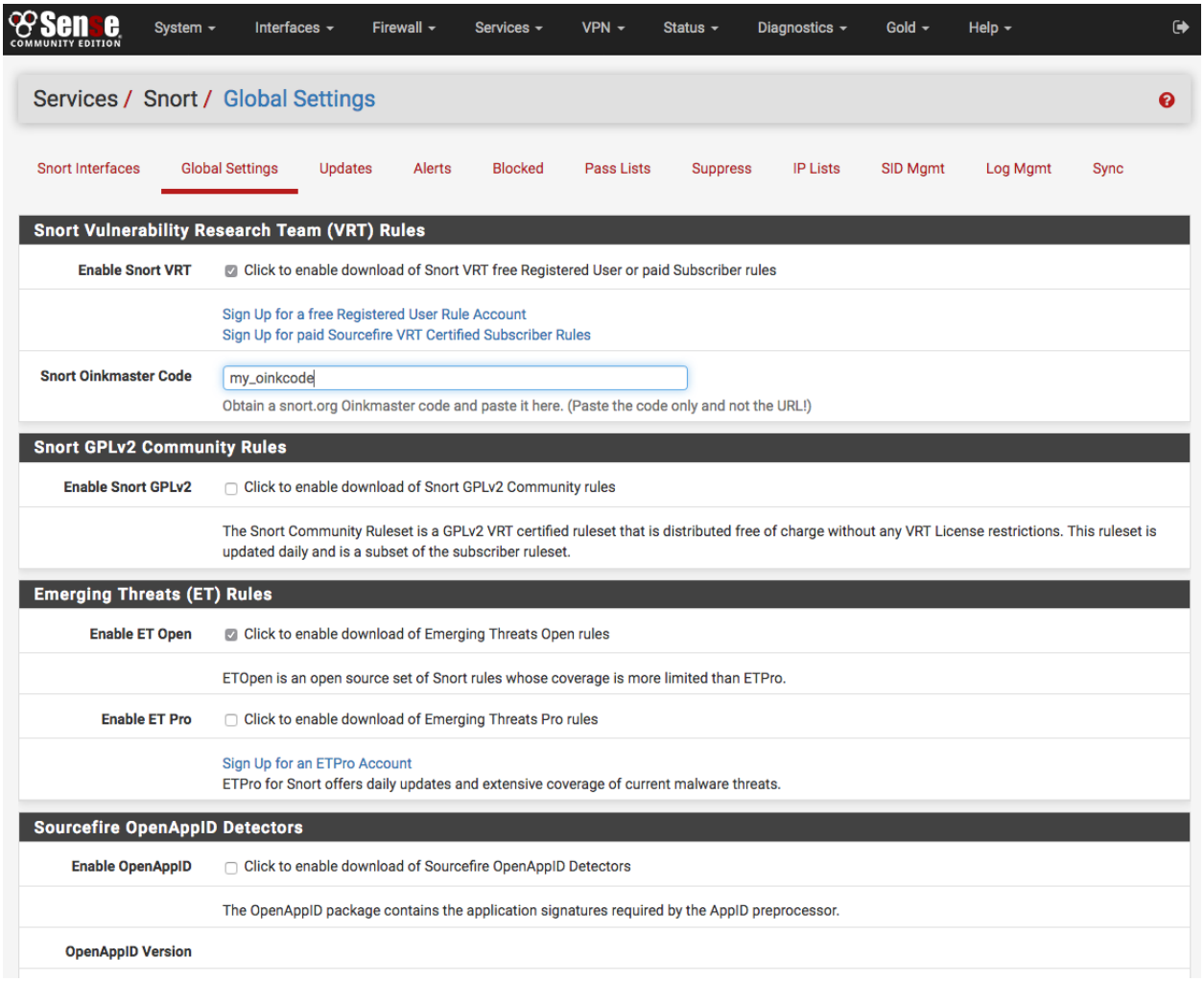


Snort interface Global Settings

This tab is used to enable rule set packages for download, configure the rules package update interval and start time, configure Snort logging directory size limits and determine whether Snort settings are saved when the package is removed from the system.

Please Choose The Type Of Rules To Download

More than one rule set may be enabled for download, but note the following caveats. If there is a paid subscription for the Snort VRT rules, then all of the Snort GPLv2 Community rules are automatically included within the file downloaded with the Snort VRT rules; therefore do not enable the GPLv2 Community rules if there is a paid-subscriber account for the Snort VRT rules. All of the Emerging Threats Open rules are included within the paid subscription for the Emerging Threats Pro rules. If the Emerging Threats Pro rules are enabled, the Emerging Threats Open rules are automatically disabled.



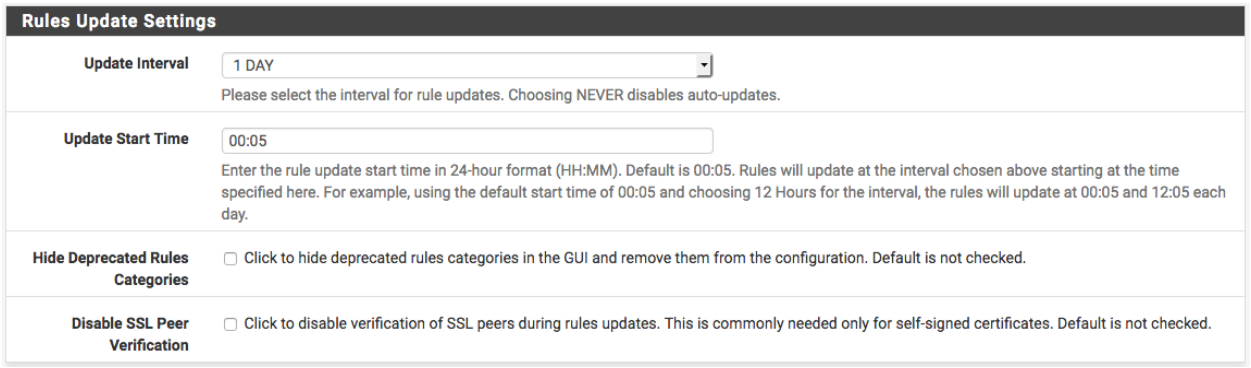
To use the Snort VRT rules package, check the **Install Snort VRT rules** checkbox and then enter the Oinkmaster code in the textbox that appears.

To use the ETPro rules package, check the box next to **ETPro** and then enter the ETPro subscription code in the textbox that appears.

Rules Update Settings

Use the **Update Interval:** drop-down selector to choose the periodicity for checking for updates to the enabled rules packages. When any value other than NEVER is selected, the **Update Start Time** textbox is available for entering a start time in 24-hour format using hours and minutes only.

In most cases every 12 hours is a good choice. The update start time can be customized if desired. Enter the time as hours and minutes in 24-hour time format. The default start time is 3 minutes past midnight local time. So with a 12-hour update interval selected, Snort will check the Snort VRT or Emerging Threats web sites at 3 minutes past midnight and 3 minutes past noon each day for any posted rule package updates.



General Settings

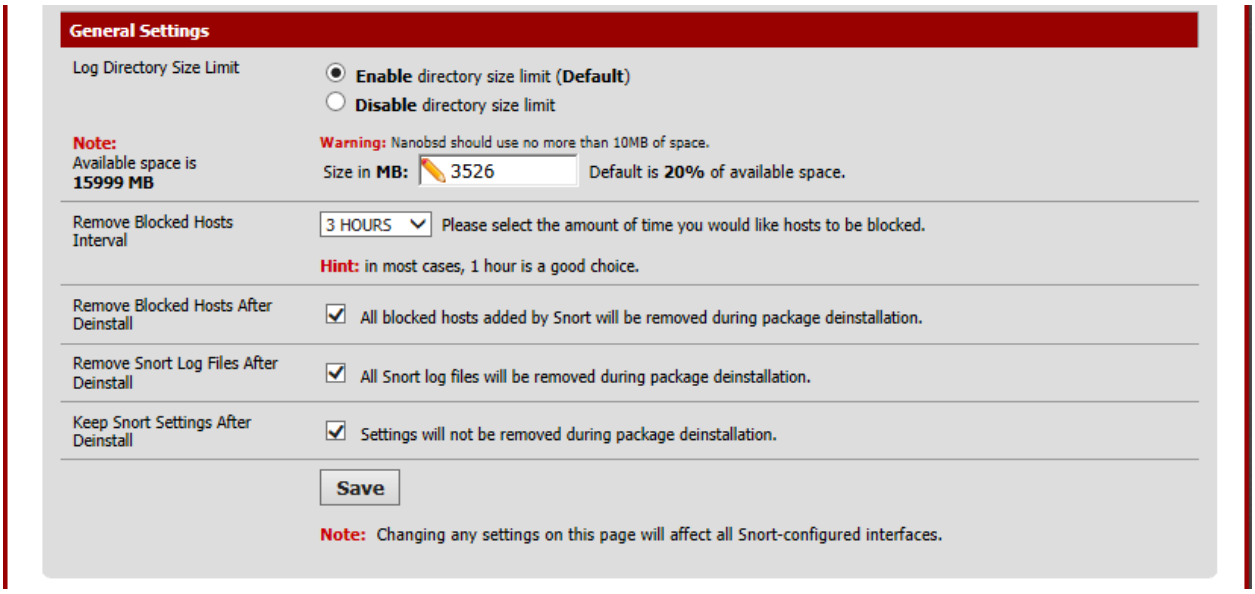
The **Log Directory Size Limit**, when enabled, sets an absolute hard upper limit on the total size of the Snort logging sub-directory in ***/var/log/snort***. This can prevent Snort from filling up the **/var** volume on the firewall. When the Snort logging directory size (the total size of all files within the Snort log directory tree) exceed the value set, all files are automatically pruned (deleted) and the Snort process is signaled to soft-restart and resynchronize logging. The default size limit is 20% of the available space on the volume. This may be overridden by setting a value in megabytes (MB) in the textbox provided.

Remove Blocked Hosts Interval: controls how long Snort-blocked IP addresses must be inactive before being cleared. Once per interval specified, Snort executes a cron job that tests all the IP addresses it has inserted into the firewall's block table for activity. IP addresses that have had no further network activity within the time specified are removed from the block table.

Remove Blocked Hosts After Deinstall: determines whether or not Snort-blocked IP addresses are automatically removed when the Snort package is uninstalled.

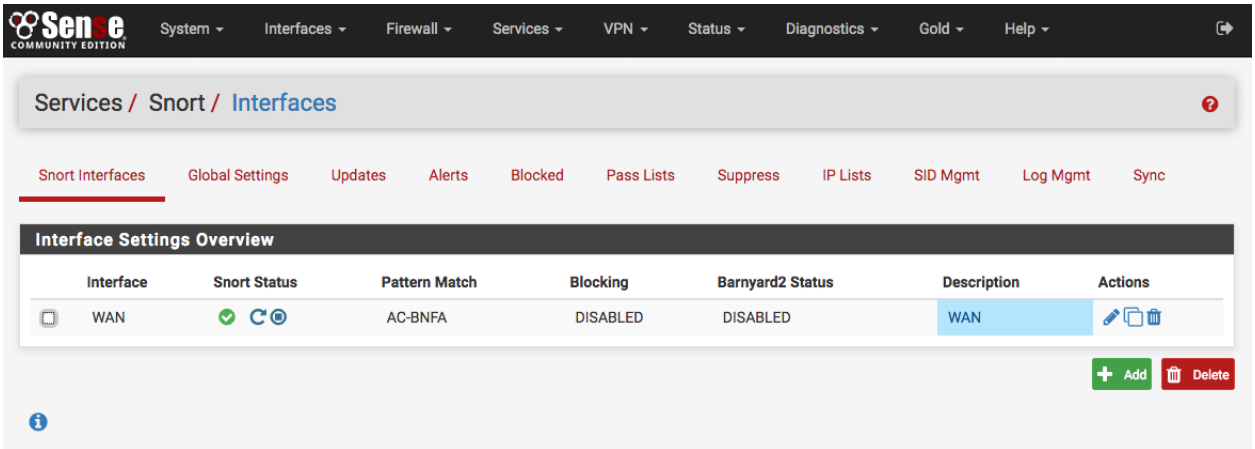
Remove Snort Log Files After Deinstall: determines whether or not log files generated by Snort are retained or removed when the Snort package is removed.

Keep Snort Settings After Deinstall: controls whether the Snort configuration is retained when the Snort package is removed.

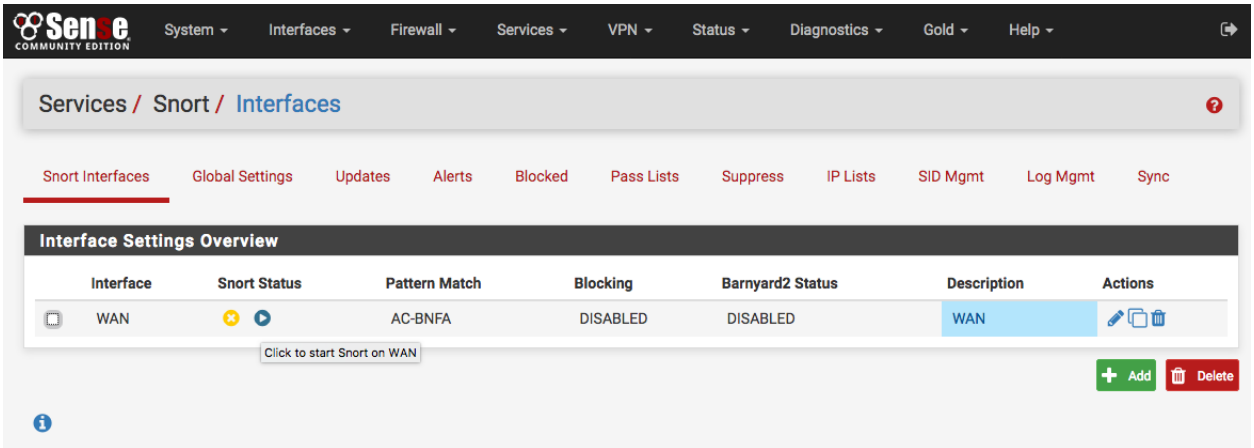


Snort Interfaces

The **Snort Interfaces** tab is where one can add, edit or delete a Snort instance from a physical network interface. A snort instance can also manually started and stopped. If *Barnyard2* is configured on an interface, it can also be started or stopped.



The green icon indicates a running Snort process for the interface. To stop a running process, click the  icon and wait for it to change to  as shown below.



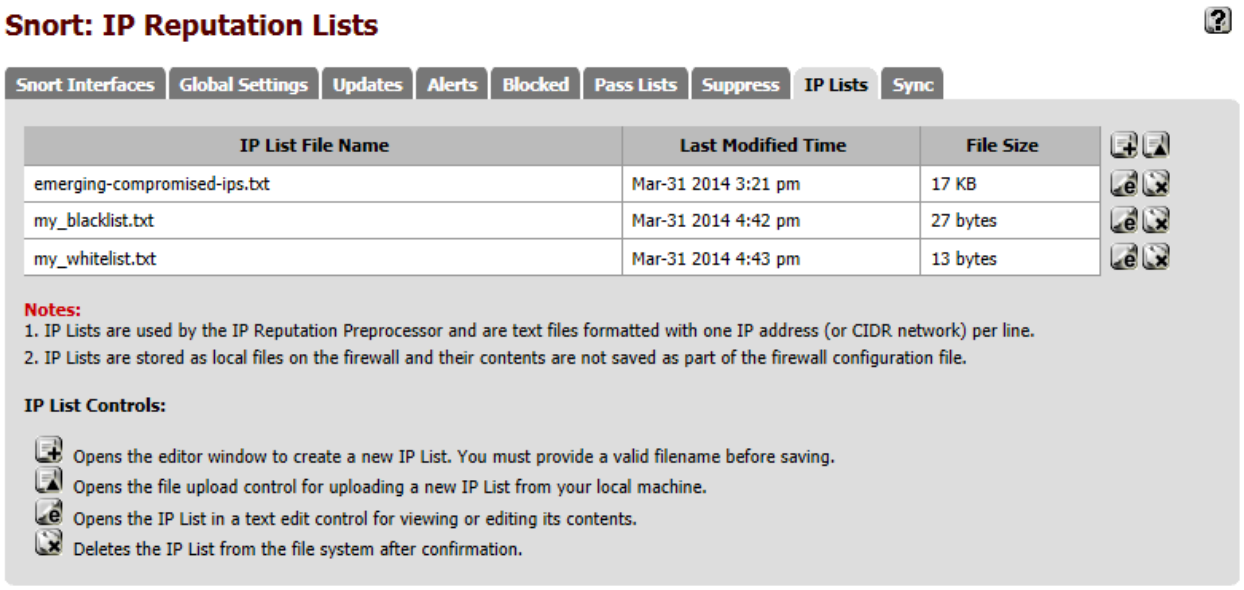
To add a new Snort configuration for an interface, click **Add**.

To edit an existing Snort configuration, click edit icon.

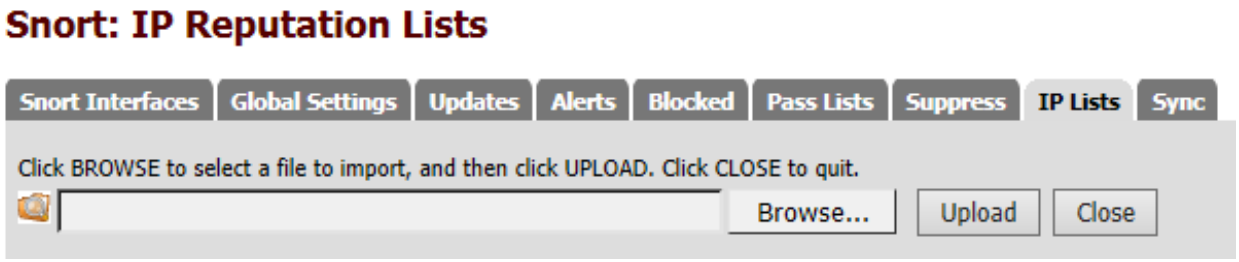
To delete an existing Snort configuration, click the checkbox on the left of the interface to select it, then click **Delete**. A prompt for confirmation will appear before deleting the chosen interface. Multiple interfaces may be selected and deleted at once.

Managing Snort IP Address Lists

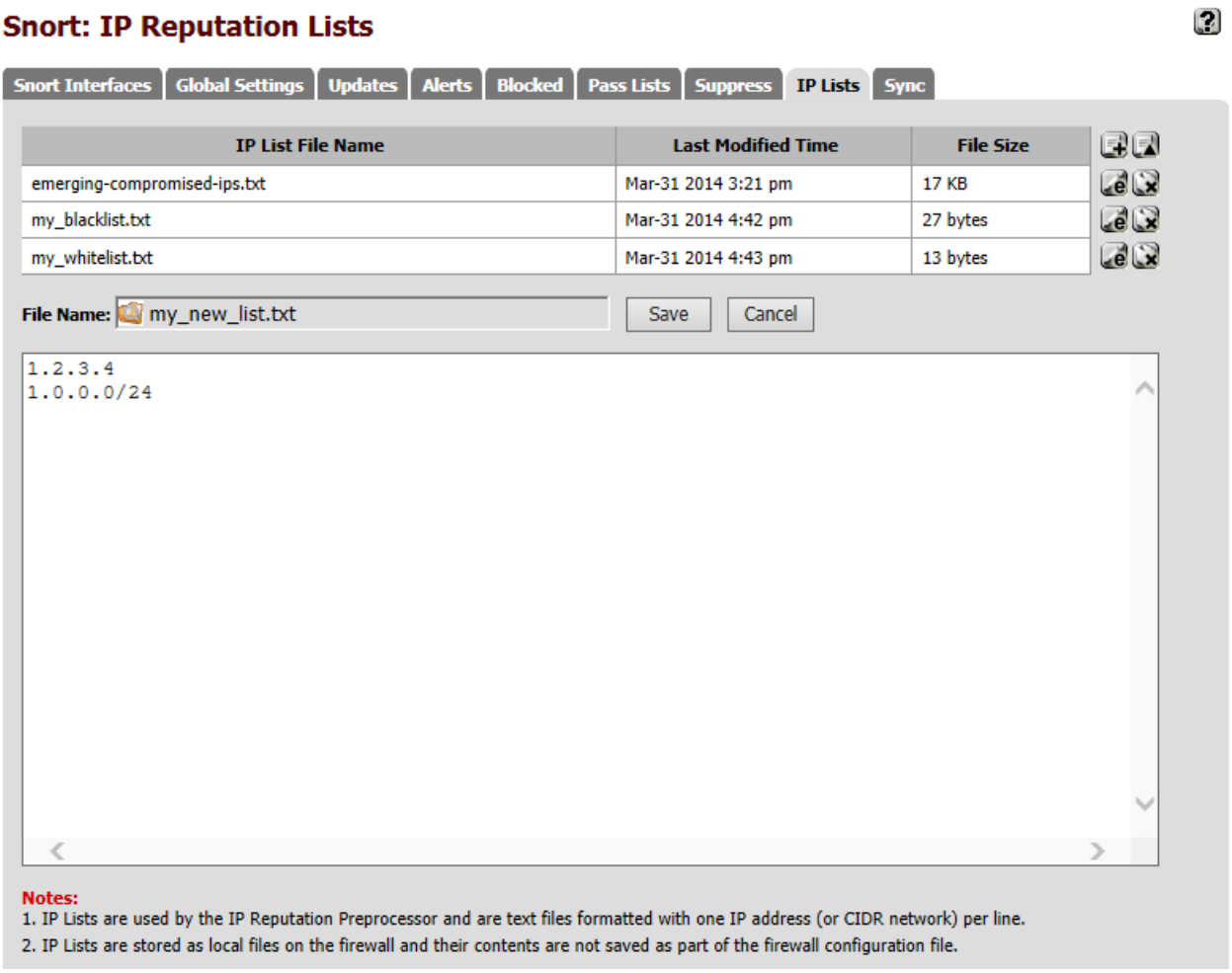
Use this tab to manage the IP lists files for the IP Reputation preprocessor. IP lists are text-format files containing one IP address or network (expressed in CIDR notation) per line.



To upload an IP list file to the firewall, click the  icon to open the file upload dialog as shown below. Browse to the file on the local machine using the **BROWSE** button, then click the **UPLOAD** button to upload the file to the firewall for use by the IP Reputation preprocessor in Snort.



To create a new IP list, click the  icon. To edit an existing IP list, click the  icon beside the list to edit. Click SAVE when finished to save changes to the list, or CANCEL to abandon any changes.



Snort IP Address Reputation Preprocessor

This tab allows configuration of the parameters specific to the IP Reputation preprocessor on the interface. It also allows the assignment of blacklist and whitelist files of IP addresses to the interface.

The available fields are:

Enable: when checked, the IP Reputation preprocessor is active on this Snort instance.

Memory Cap: sets the amount of system memory in megabytes (MB) to reserve for storage of the IP lists associated with this preprocessor. The default is 500 MB and should be sufficient for most installations.

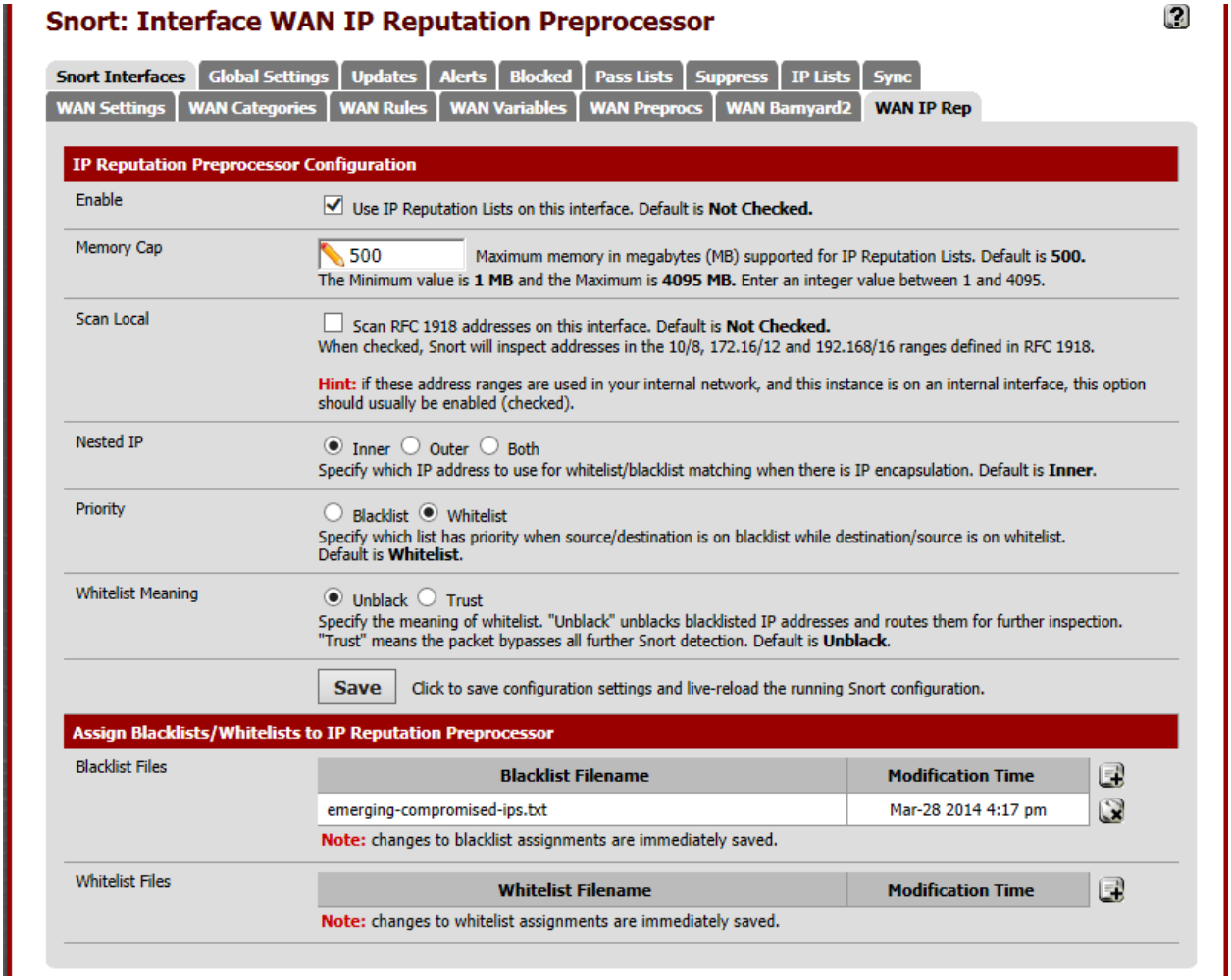
Scan Local: when checked, Snort will include RFC 1918 IP addresses ranges when comparing IP addresses to the blacklists and whitelists. If an RFC 1918 IP addresses is in the whitelist files, or some are blacklist files, then this option should be enabled. The default is disabled.

Nested IP: this tells Snort which IP address to compare to the IP lists in the whitelist and blacklist files when there is IP encapsulation. The default is **Inner**.

Priority: instructs Snort which IP list has priority when the source and destination IP addresses of a packet are each on separate IP lists. For example, if the source IP address is on a blacklist while the destination IP address is on a whitelist, this option tells Snort whether to block the traffic if blacklist has priority, or pass the traffic if whitelist has priority.

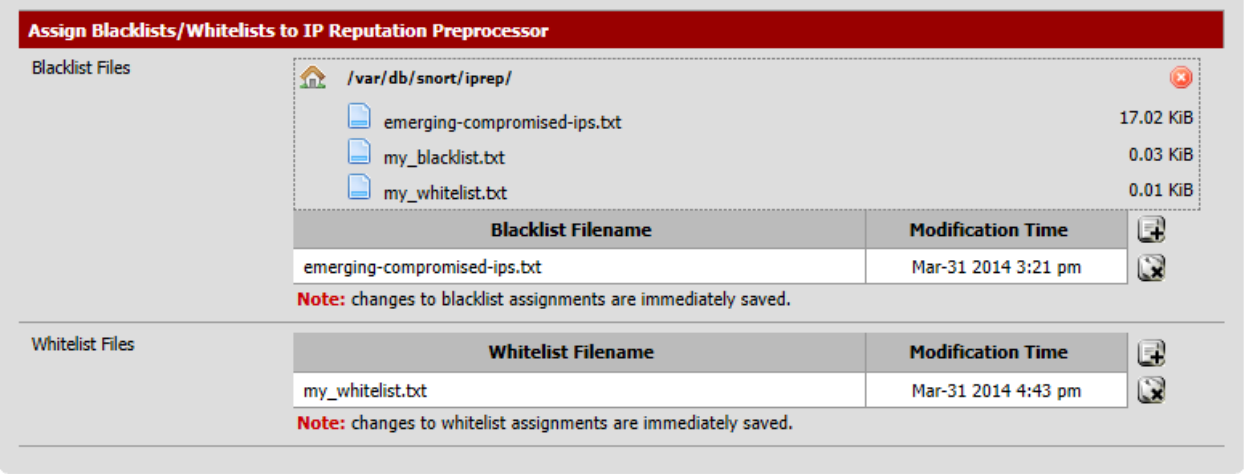
Whitelist Meaning: this tells Snort what action to take with whitelisted IP addresses. The two options are **Un-black** and **Trust**. When set to **Un-black**, a blacklisted IP which is listed in the whitelist is not immediately blocked. Instead it is routed through the Snort detection engine for normal inspection. If it generates no alerts, the traffic is allowed. If the inspection results in a Snort alert for the traffic, it will be blocked.

When set to **Trust**, any IP address on the whitelist (including any that may also be on a blacklist) is immediately allowed to pass with no further inspection. Caution should be exercised when using the Trust mode of operation to insure the IP addresses on the whitelist are in fact trustworthy.



The  and  icons at the bottom of the page are used to assign or remove blacklist and whitelist files to or from the interface.

Click the  icon to open a file selection dialog. Choose an IP list file from the list by clicking on the name.



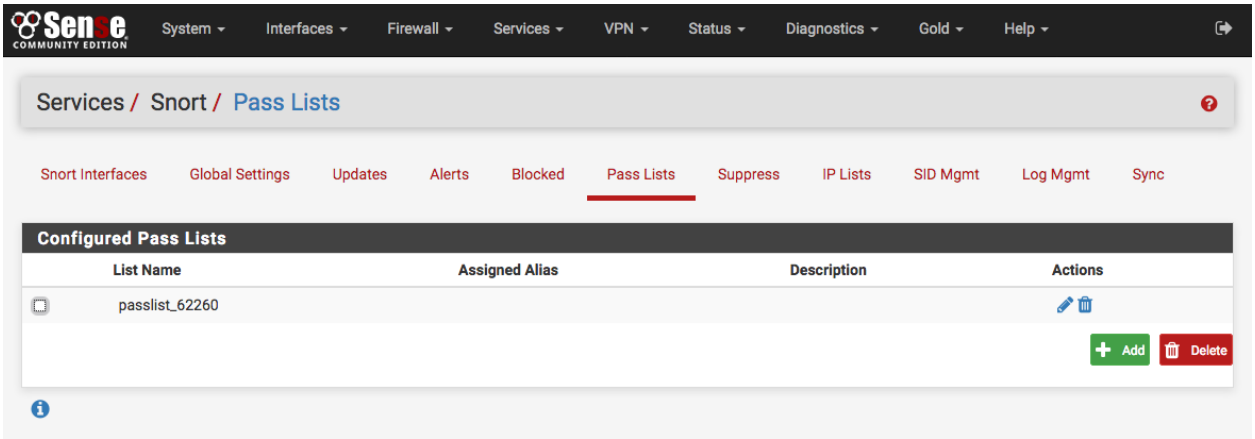
Snort IP Address Reputation

Coming soon...

Snort Pass Lists

Pass Lists are lists of IP addresses that Snort should never block. Pass lists can be created and managed on the **Pass Lists** tab. When an IP address is listed on a Pass List, Snort will never insert a block on that address even when malicious traffic is detected.

To create a new Pass List, click the  icon. To edit an existing Pass List, click the  icon. To delete a Pass List, click the  icon. Note that a Pass List cannot be deleted if it is currently assigned to one or more Snort interfaces.

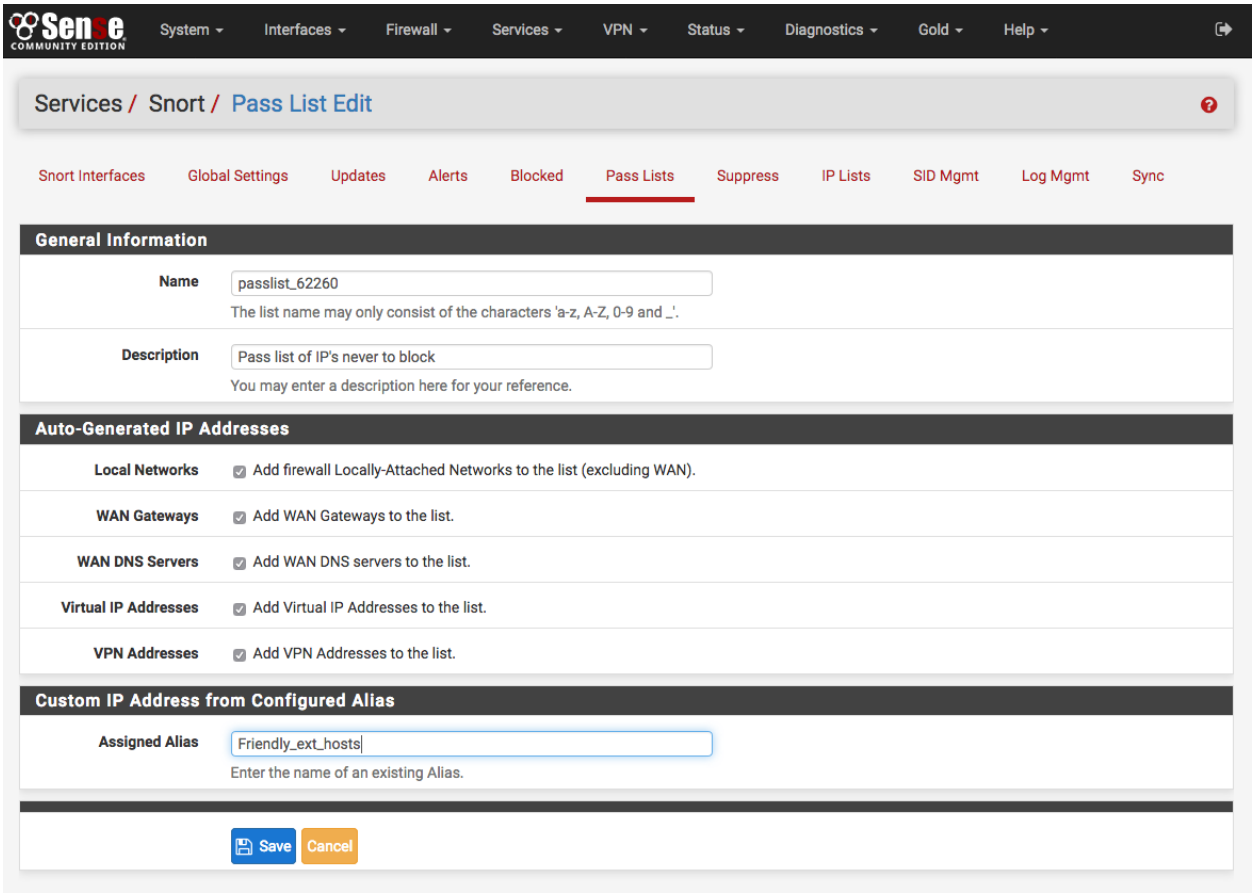


A default Pass List is automatically generated by Snort for every interface, and this default list is used when no other list is specified. Assign Pass Lists to an interface on the **Interface Settings** tab.

Customized Pass Lists can be created and then assigned to an interface. This might be needed when trusted external hosts are needed that are not located on networks directly connected to the firewall. To add external hosts in this manner, first create an Alias under **Firewall > Aliases** and then assign that alias to the **Assigned Aliases:** field. In the

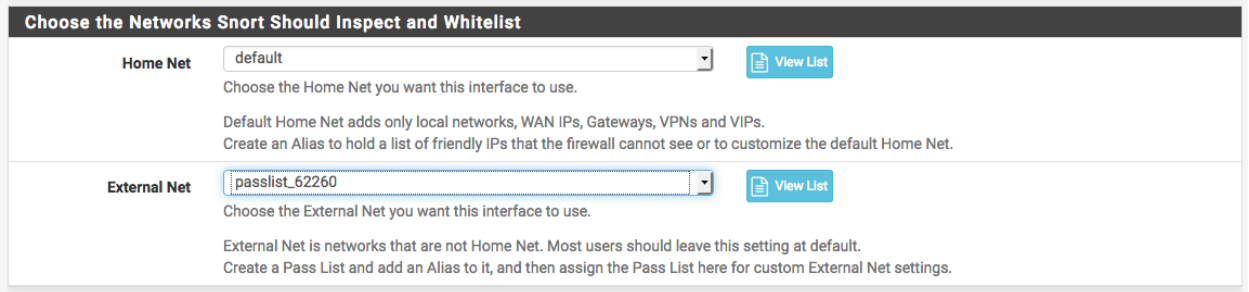
example shown below, the alias "*Friendly_ext_hosts*" has been assigned. This alias would contain the IP addresses of the trusted external hosts.

When creating a custom Pass List, leave all the auto-generated IP addresses checked in the **Add auto-generated IP addresses** section. Not selecting the checkboxes in this section can lead to blocking of critical addresses including the firewall interfaces themselves. This could result in being locked out of the firewall over the network! Only uncheck boxes in this section when a valid need is present.



Click the **ALIASES** button to open a window showing previously defined aliases for selection. Remember to click **SAVE** to save changes.

Note: Remember that simply creating a Pass List is only the first step! Go to the **Interface Settings** tab for the Snort interface and assign the newly created Pass List as shown below. After assigning and saving the new Pass List, restart Snort on the affected interface to pick up the change.

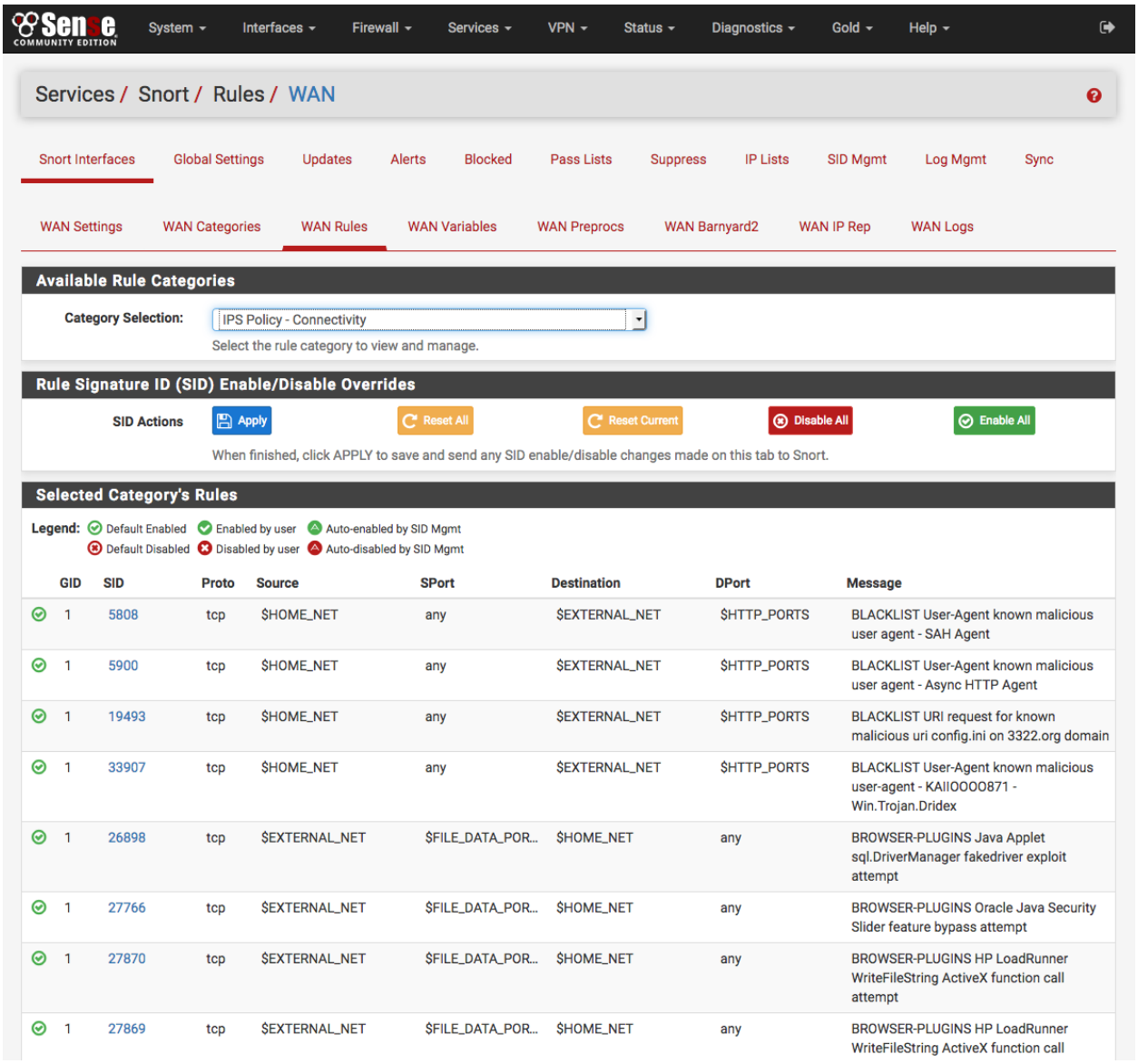


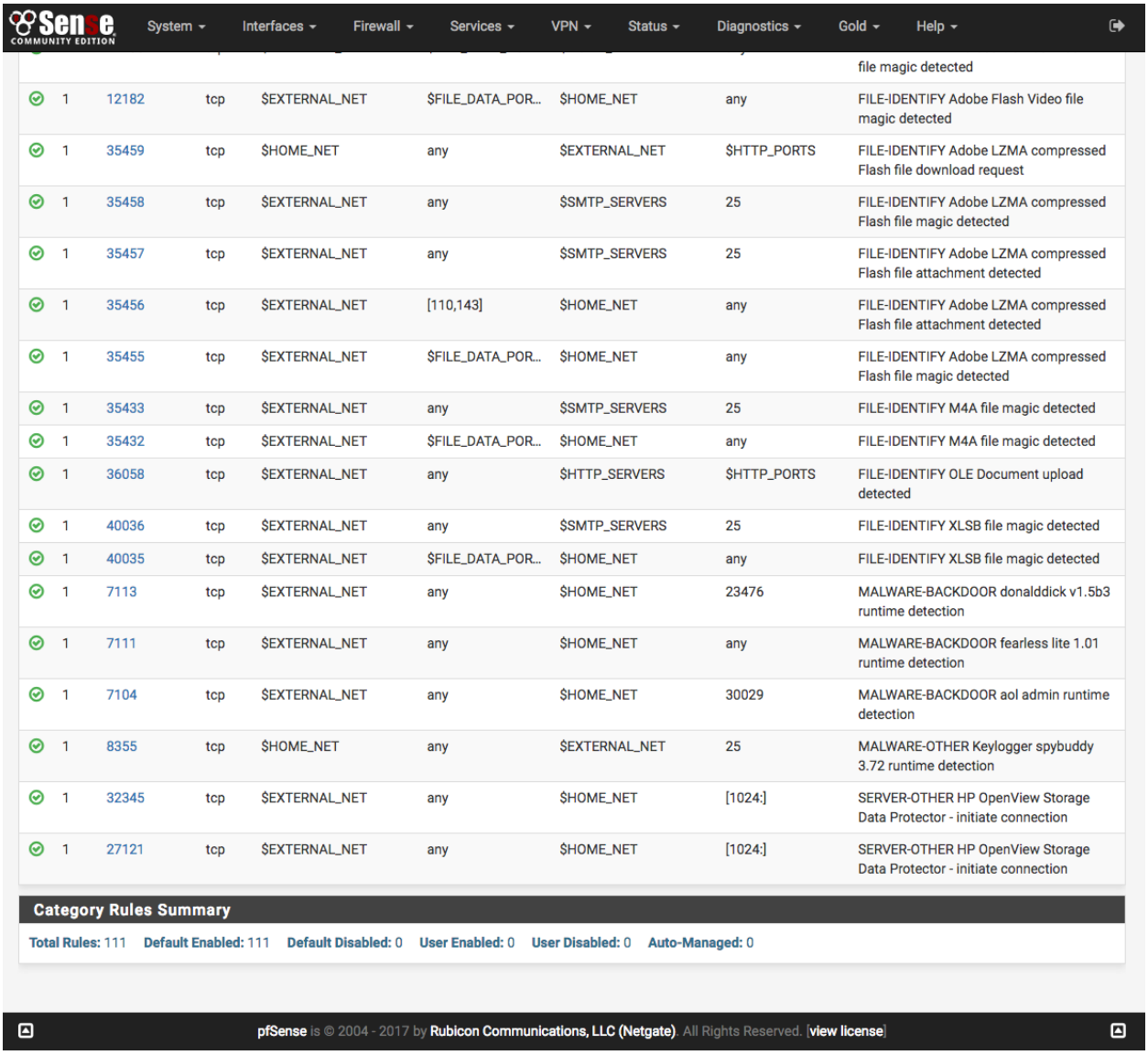
Snort Rules

Rules

Use the **Rules** tab for the interface to configure individual rules in the enabled categories. Generally this page is only used to disable particular rules that may be generating too many false positives in a network environment. Be sure they are in fact truly false positives before taking the step of disabling a Snort rule!

Select a rules category from the **Category:** drop-down to view all the assigned rules. Click the or icon at the far-left of a row to toggle the rule's state from enabled to disabled, or click or to toggle from disabled to enabled. The icon will change to indicate the state of the rule. At the top of the rule list is a legend showing the icons used to indicate the current state of a rule.



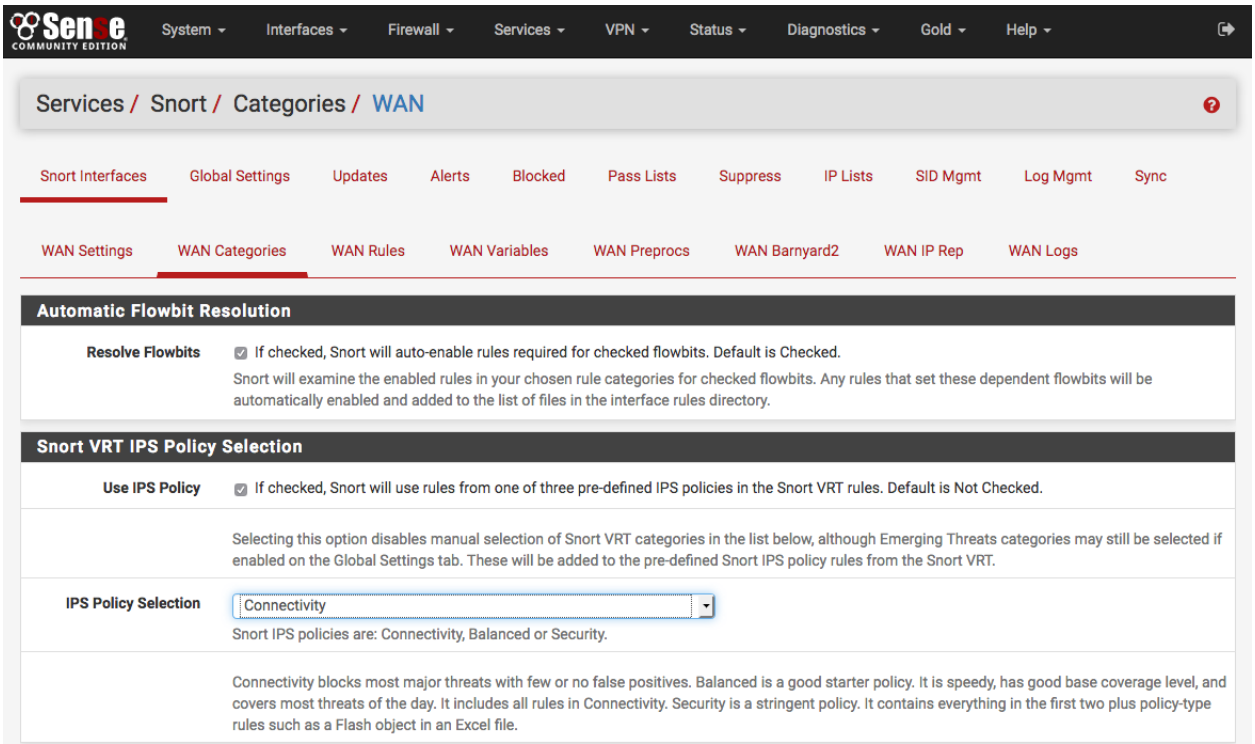


Snort Rulesets

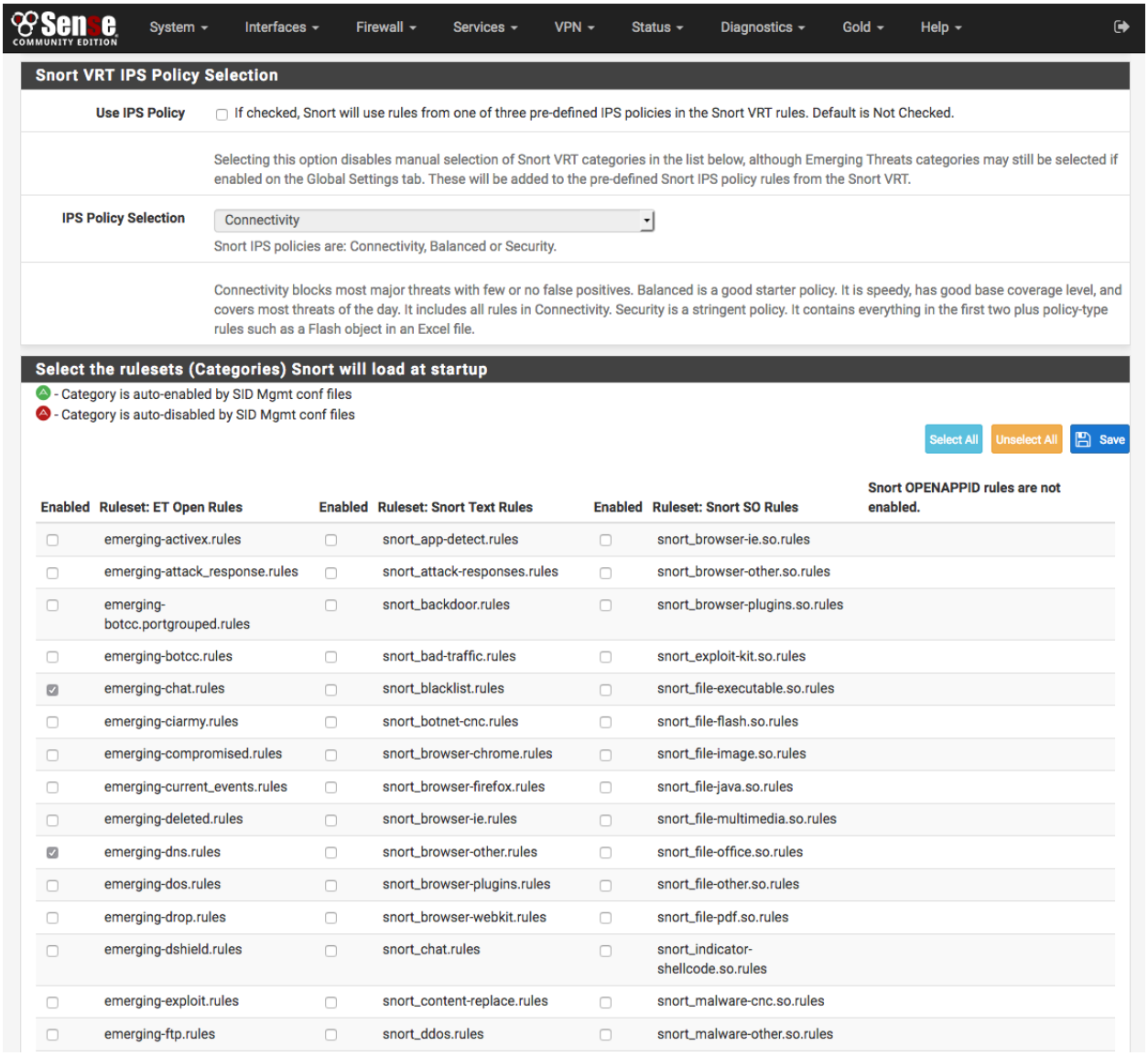
Categories

If a Snort VRT Oinkmaster code has been obtained (either free registered user or the paid subscription), and the Snort VRT rules have been enabled, and the Oinkmaster code has been entered on the Global Settings tab then the option of choosing from among three pre-configured IPS policies is available. These greatly simplify the process of choosing enforcing rules for Snort to use when inspecting traffic. The IPS policies are only available when the Snort VRT rules are enabled.

The three Snort VRT IPS Policies are: (1) Connectivity, (2) Balanced and (3) Security. These are listed in order of increasing security. However, resist the temptation to immediately jump to the most secure “Security” policy if new to using Snort. False positives can frequently occur with the more secure policies, and careful tuning by an experienced administrator may be required. So if new to Snort, then using the less restrictive “Connectivity” policy in non-blocking mode is recommended as a starting point. Once experience with Snort has been gained in the network environment, blocking mode can be enabled and then move up to more restrictive IPS policies.



If the Snort VRT rules are not enabled, or to use any of the other rule packages, then make the rule category selections by checking the checkboxes beside the rule categories to use.

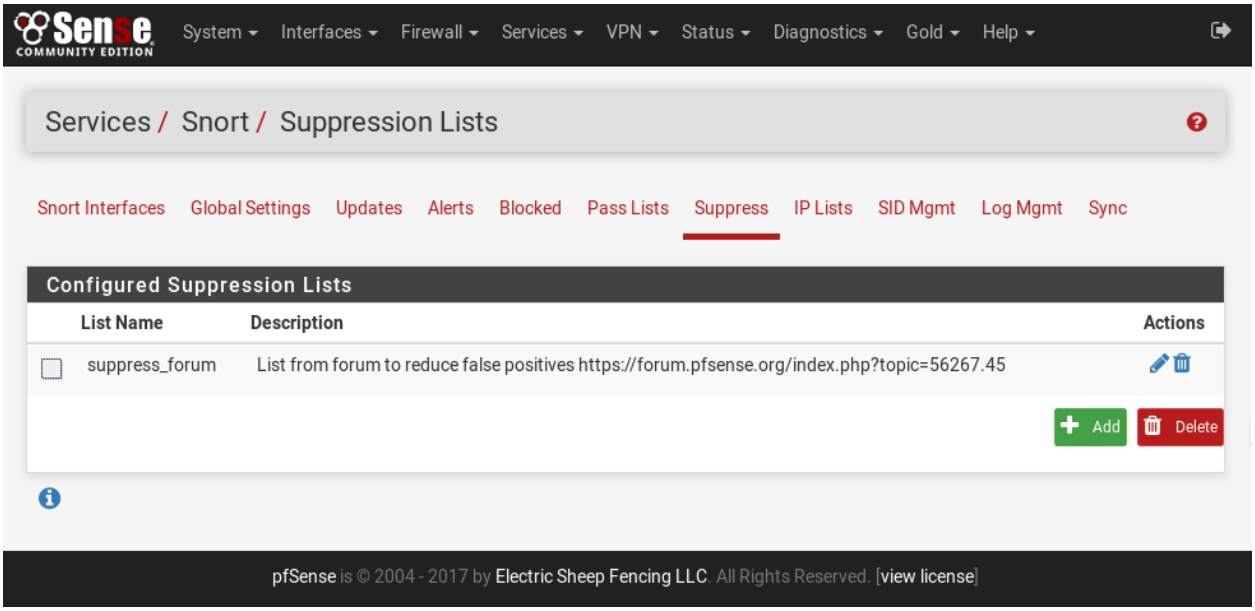


Be sure to click **SAVE** when finished to save the selection and build the rules file for Snort to use.

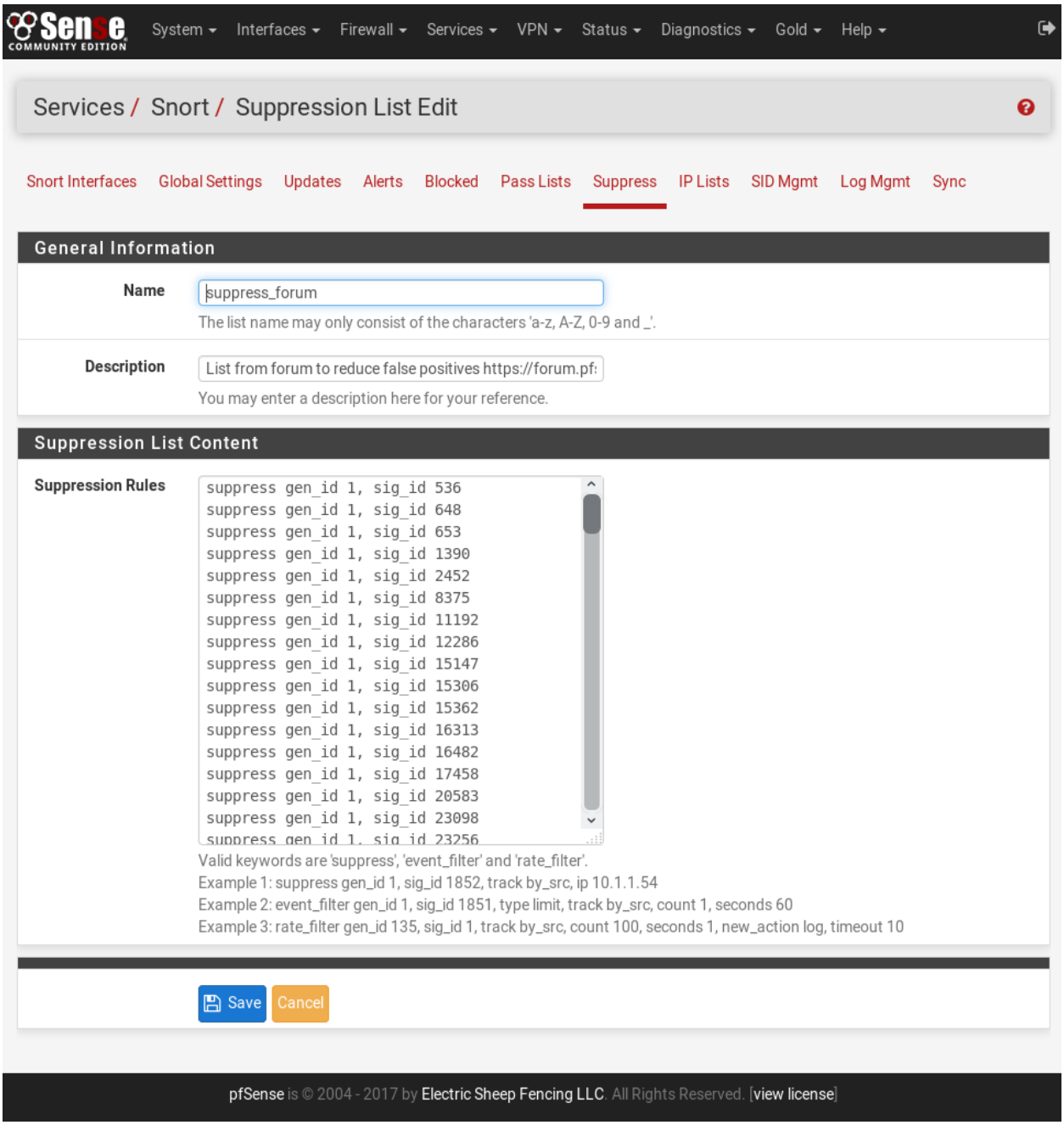
Snort Suppression Lists

Alert Thresholding and Suppression

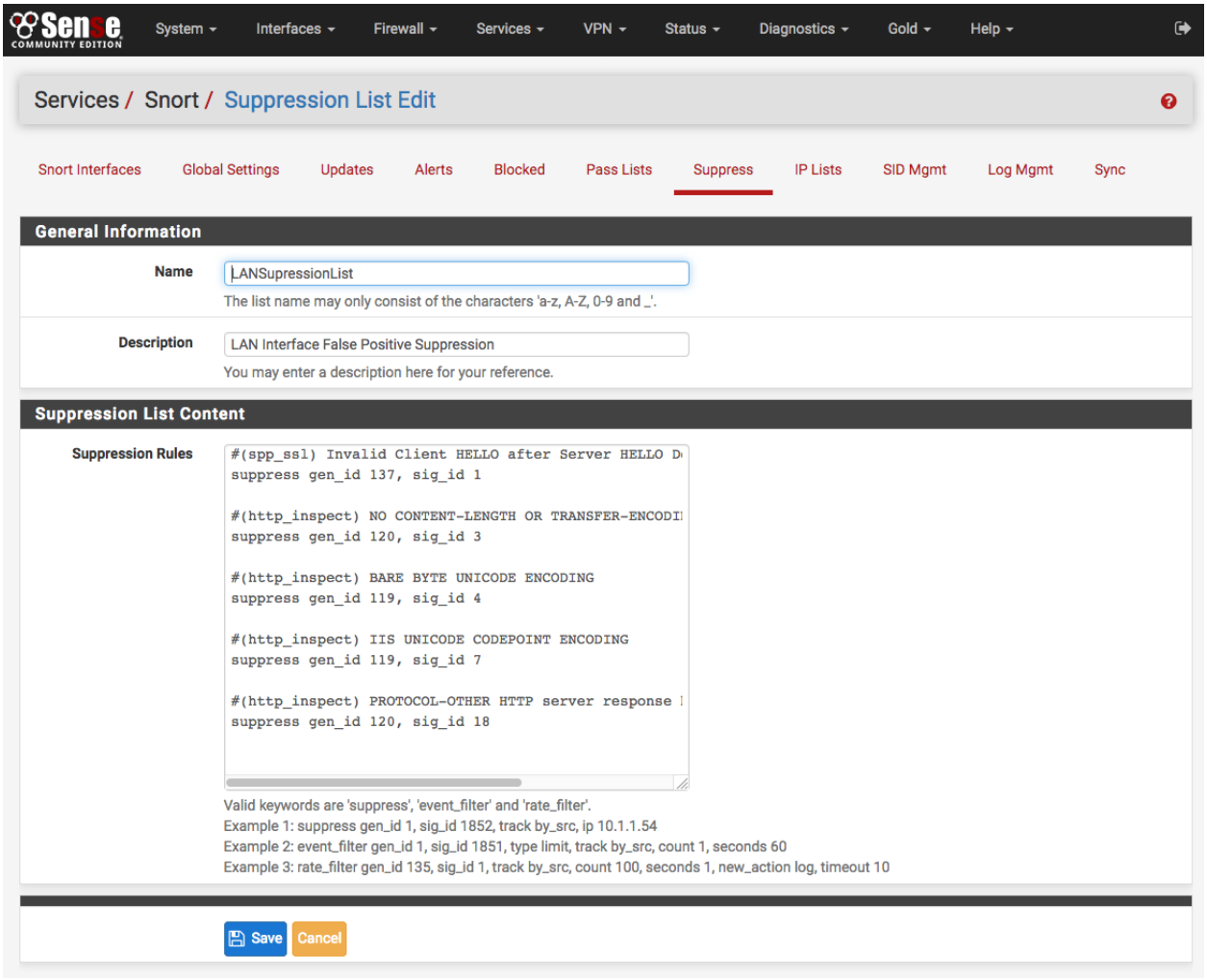
Suppression Lists allow control over the alerts generated by Snort rules. When an alert is suppressed, then Snort no longer logs an alert entry (or blocks the IP address if block offenders is enabled) when a particular rule fires. Snort still inspects all network traffic against the rule, but even when traffic matches the rule signature, no alert will be generated. This is different from disabling a rule. When a rule is disabled, Snort no longer tries to match it to any network traffic. Suppressing a rule might be done in lieu of disabling the rule to stop alerts based on either the source or destination IP. For example, to suppress the alert when traffic from a particular trusted IP address is the source. If any other IP is the source or destination of the traffic, the rule may still be desired. To eliminate all alerts from the rule, then it is more efficient to simply disable the rule rather than to suppress it. Disabling the rule will remove it from the list of match rules in Snort and therefore makes for less work Snort has to do.



On the Suppress List Edit page, suppress lists may be manually added or edited. It is usually easier and faster to add suppress list entries by clicking the  icons shown with the alert entries on the **Alerts** tab. Remember to click the **SAVE** button to save changes when manually editing Suppress List entries.



Lists with comments are easier to manipulate and fine tune. Neither screen shot shows IP address suffix in a suppress entry.



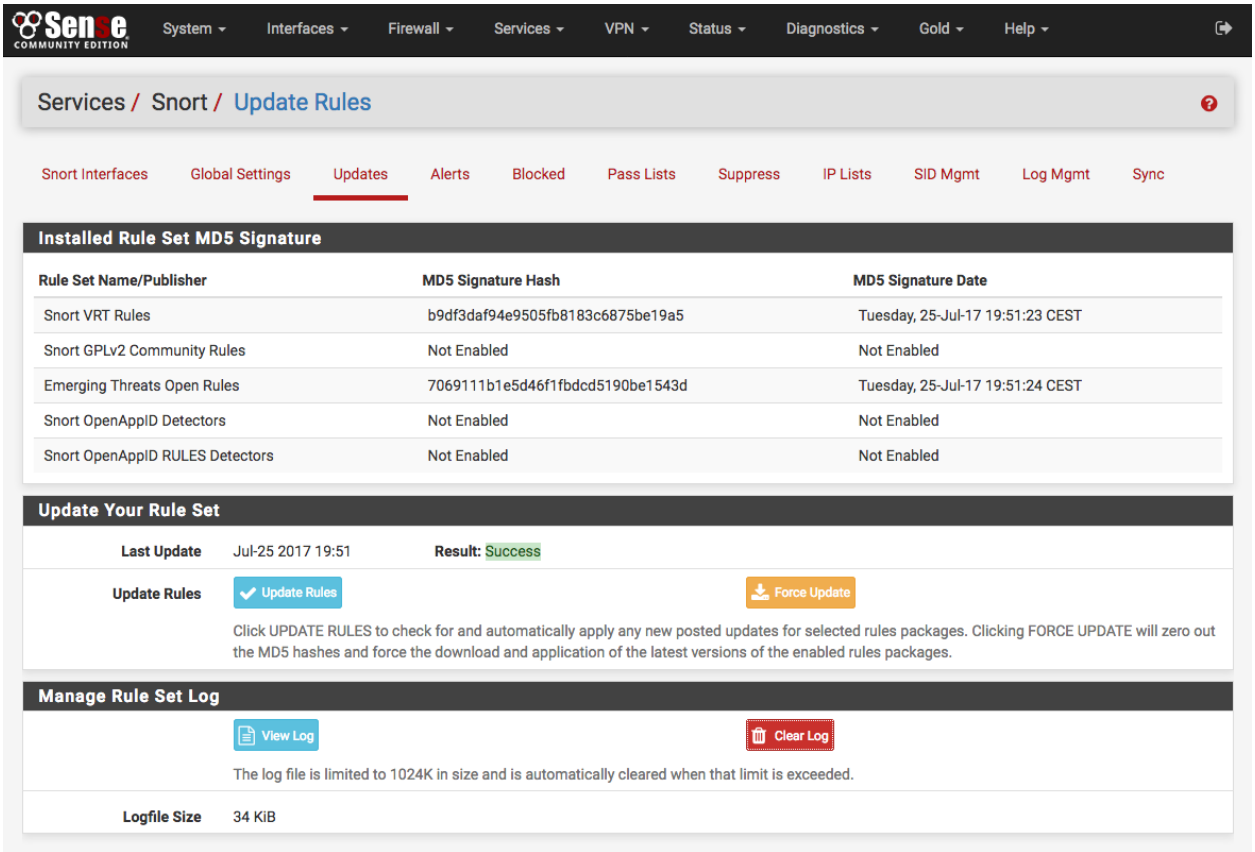
Updating Snort

Update the rules

The **Updates** tab is used to check the status of downloaded rules packages and to download new updates. The table shows the available rule packages and their current status (not enabled, not downloaded, or a valid MD5 checksum and date).

Click on the **Update Rules** button to download the latest rule package updates. If there is a newer set of packaged rules on the vendor web site, it will be downloaded and installed. The determination is made by comparing the MD5 of the local file with that of the remote file on the vendor web site. If there is a mismatch, a new file is downloaded. The **FORCE** button can be used to force download of the rule packages from the vendor web site no matter how the MD5 hash tests out.

In the screenshot below, the Snort VRT and Emerging Threats Open rule packages have been successfully downloaded. The calculated MD5 hash and the file download date and time are shown. Also note the last update time and result are shown in the center of the page.



See also:

Troubleshooting Snort Rule Updates

Tip: For more information or to get help, check out the IDS/IPS category on the Netgate forum.

29.24 Stunnel package

The stunnel program is designed to work as an SSL encryption wrapper between remote client and local (inetd-startable) or remote servers. It can be used to add SSL functionality to commonly used inetd daemons like POP2, POP3, and IMAP servers without any changes in the program's code.

It will negotiate an SSL connection using the **OpenSSL** or **SSLeay** libraries. It calls the underlying crypto libraries, allowing stunnel to support whatever cryptographic algorithms were compiled into the crypto package.

Note: The pfSense® package implements only a subset of the configuration options available in stunnel. For more advanced configurations, please consider configuring stunnel manually on the pfSense host, run it in a dedicated jail, or on a different system.

The package has two configuration screens (tabs):

- Tunnel definitions
- Certificates

29.24.1 Tunnels

For each tunnel, the following options are available:

- Listening socket IP address and port.
- Certificate to use for the listening socket.
- Target IP address and port.
- IP address to bind to when connecting to the target.

If no certificate is specified for a tunnel, the default certificate will be used. This is a self-signed certificate which is generated upon package (re)installation, and is not suited for production use.

29.24.2 Certificates

Certificates are managed in the simplest possible way, by requiring the user to provide RSA key and certificates/chains in PEM format. The **Certificates** tab will list the configured certificates along with status information, indicating whether the certificate is valid, will expire soon, or is already expired. A sanity check is also performed to make sure the key and certificate matches.

Note that for private certificates and certain commercial ones (Extended Validation), a complete certificate chain may be required. This is to ensure that the client is able to verify the certificate validity. A chain should be built in the following way:

1. Root certificate of the certificate issuer/CA
2. Any intermediate certificates between the root and the server certificate
3. Server certificate

See also:

Refer to the stunnel documentation for more information on how to format a certificate chain.

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.25 Sudo Package

The sudo package configures basic rules for allowing unprivileged users (i.e. anyone but `root`/`admin`) to run commands as `root` or another user/group in the shell.

Once the `sudo` package is installed, it is located at **System > sudo** in the GUI.

29.25.1 Sudo Settings

The package allows multiple entries for privileges. To add a new entry, click  **Add**, and fill in the settings:

User/Group The user or group name to which this privilege is being granted.

The list includes users and groups defined in the GUI as well as those from the operating system (e.g. daemon users and groups added by packages).

Run As The user or group name under which the command will be run.

In most cases this is `root`, so that users may run commands as `root` without knowing the `root`/`admin` credentials.

No Password Controls whether or not the user is not prompted for their own password when executing commands using `sudo`.

This is unset by default, so users are prompted for their password when running `sudo`. `sudo` caches credentials in a login session for at least five minutes after each execution of `sudo` so that the user is not prompted on each attempt. Users can execute `sudo` without reauthenticating inside that time frame, but if they stop for five minutes they will be prompted again on the next run.

When set, the user is not prompted for their password when running `sudo`. This is less secure, but more convenient. If `sudo` is invoked non-interactively, such as from a cron script, this is required as there is no way for a user to enter their credentials.

Command List A list of commands the **User/Group** can run.

See also:

More information on the full command options may be found in the sudoers manual.

By default the command is `ALL` meaning the user can run any commands. Leaving the commands field blank assumes `ALL`.

A comma-separated list of one or more commands can be supplied to limit the user to individual binaries. Full paths to commands are required by `sudo` to ensure the user is properly restricted to specific binaries or scripts.

If parameters are specified after a command, they will be required. To disallow running a command with parameters, add `""` after the command.

Custom Configuration This option controls whether or not `sudo` will read additional configuration files from `/usr/local/etc/sudoers.d`.

> **Warning:** Including custom configuration files allows options to be set which are not supported by the GUI, but these files can be a potential security risk and they are not included in backups.

The setting can be one of:

Do Not Include `sudo` will **not** include additional configuration files.

Include at Start `sudo` will include additional configuration files **before** the GUI settings.

Include at End `sudo` will include additional configuration files **after** the GUI settings.

29.25.2 Sudo Examples

Example 1

Allow `bob` to run `ping` commands only as `root` without a password:

User/Group *User*: `bob`

Run As *User*: `root`

No Password checked

Commands `/sbin/ping`

Example 2

Allow anyone in the admins group to run all commands as any user, but prompted for a password:

User/Group *Group*: `admins`

Run As *User*: *ALL Users*

No Password Unchecked

Commands `ALL`

Command Examples

These examples demonstrate how to specify commands in various ways.

- Run `ping` with any parameters:

```
/sbin/ping
```

- Run `ping` only to `192.168.1.2`:

```
/sbin/ping 192.168.1.2
```

- Run command `blah` without any parameters:

```
/usr/local/bin/blah ""
```

- Run `ping` and `traceroute` and their IPv6 variants with any parameters:

```
/sbin/ping, /sbin/ping6, /usr/sbin/traceroute, /usr/sbin/traceroute6
```

Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

29.26 Status Traffic Totals

This package displays different ways to view the traffic usage generated by the network traffic monitoring tool vnStat.

See also:

Similar tools are available for pfSense bandwith monitoring *How can I monitor bandwidth usage*

29.26.1 Notes

Every NIC is added on install. So if a NIC is added (or removed) on the pfSense® box, remove the package and install again. If the pfSense router has data for a NIC vnStat will report the data even if the NIC has been removed.

A reinstall of the package will not change this as the core pfSense system has data pertaining to the non existent data and thus other packages such as vnstat2 will report the data it has or has found

29.26.2 Known Issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

29.26.3 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

CHAPTER

THIRTY

VIRTUALIZATION

pfSense® software supports a variety of Type-1 (bare metal/native) and Type-2 (hosted) virtualization environments, such as VMware (vSphere, Fusion or Workstation), Proxmox VE, VirtualBox, Xen, KVM, Hyper-V and so on.

Warning: We recommend using Type-1 hypervisors for production use. Type-2 hypervisors such as VirtualBox or VMware Workstation work fine for testing, but avoid using them for production roles where possible.

Set up and install is straightforward and similar to set up on a physical machine using the ISO image.

30.1 VirtIO Driver Support

The FreeBSD kernel used by pfSense® software includes VirtIO drivers built into the kernel. No special action is necessary to enable the drivers.

30.1.1 Disable Hardware Checksum Offloading

With the current state of VirtIO network drivers in FreeBSD, it is necessary to check **Disable hardware checksum offload** under **System > Advanced** on the **Networking** tab *and to manually reboot pfSense after saving the setting, even though there is no prompt instructing to do so* to be able to reach systems (at least other VM guests, possibly others) protected by pfSense software **directly from the VM host**.

The issue is most likely caused by FreeBSD Bug 165059.

Hardware checksums and other NIC offloading features like TSO may also need to be disabled on the hypervisor system in addition to the pfSense VM.

30.2 Guides

- *Virtualizing pfSense with VMware vSphere / ESXi*
- *Virtualizing pfSense with Hyper-V*
- *Virtualizing with Proxmox® VE*

CHAPTER

THIRTYONE

WIRELESS

31.1 Should I use a pfSense appliance as my access point

Historically, the access point functionality in FreeBSD has suffered from serious compatibility or performance problems with some wireless clients. Over time this has improved significantly. pfSense® access points are used in various locations with no trouble. It is used with various clients such as MacBook Pro, Apple AirTunes, Mac mini, iPod Touch, Android devices, Palm, various Windows laptops, Xbox 360, and FreeBSD clients and it works very reliably.

There is the possibility of finding incompatible devices with any access point. FreeBSD is no exception and it can be more common with FreeBSD than other access points. Using a pfSense device as an access point can work quite well with the right card and configuration.

In general, we still recommend *Using an External Wireless Access Point*, especially if 802.11n or 802.11ac are required. Placing that burden on an external device and allowing pfSense software to focus on the firewall/routing/NAT/etc can be simpler in the long run. An access point may still be connected to a dedicated interface or VLAN for isolation purposes.

31.1.1 Incompatible wireless clients

There are no known incompatible devices at this time.

31.2 Recommended Wireless Hardware

A variety of wireless cards are supported in FreeBSD 12.2-STABLE@f4d0bc6aa6b, and pfSense® includes support for every card supported by FreeBSD. Some have better support than others. Most pfSense developers work with Atheros hardware, so it tends to be the most recommended hardware. Many have success with other cards as well, and Ralink is another popular choice. Other cards may be supported, but do not support all available features. In particular, some Intel cards can be used in infrastructure mode as clients but cannot run in access point mode due to limitations of the hardware itself.

31.2.1 Wireless cards from big name vendors

Linksys, D-Link, Netgear and other major manufacturers commonly change the chipset used in their wireless cards without changing the model number. There is no way to ensure a specific model card from these vendors will be compatible because there is no reliable way of knowing which "minor" card revision and chip a package contains. While one revision of a particular model may be compatible and work well, another card of the same model may be incompatible. For this reason, we recommend avoiding cards from the major manufacturers. If a card is already on hand, it is worth trying to see if it is compatible. Be wary when purchasing because even if the "same" model worked for someone else, a new purchase may result in a completely different piece of hardware that is incompatible.

31.2.2 Status of 802.11n Support

pfSense 2.5.2-RELEASE is based on FreeBSD 12.2-STABLE@f4d0bc6aa6b which has support for 802.11n on certain hardware such as those based on the Atheros AR9280 and AR9220 chipsets. We have tested cards using those chipsets and they work well. Some other non-Atheros cards are documented by FreeBSD to work on 802.11n, specifically, *mwl(4)* and *iwn(4)*. These may work using the 802.11n standard but experiences with 802.11n speeds may vary.

The FreeBSD Wiki Article for 802.11n Support contains the most up-to-date information about supported chipsets and drivers that work with 802.11n.

31.2.3 Status of 802.11ac Support

Currently, there is no support for 802.11ac in FreeBSD nor in pfSense.

31.2.4 Radio Frequencies and Dual Band Support

Some cards have support for 2.4GHz and 5GHz bands, such as the Atheros AR9280, but only one band may be used at a time. Currently there are no cards supported and working in FreeBSD that will operate in both bands concurrently. Using two separate cards in one unit is not desirable as their radios may interfere. In cases that require dual or multiple band support, we strongly recommend an external AP.

31.2.5 Wireless drivers included in pfSense

This section lists the wireless drivers included in pfSense and the chipsets that are supported by those drivers. This information was derived from the FreeBSD man pages for the drivers in question. Drivers in FreeBSD are referred to by their driver name, followed by (4), such as *ath(4)*. The (4) refers to the kernel interfaces section of the man page collection, in this case specifying a network driver. The drivers are listed in order of frequency of use with pfSense, based on reports from users.

For more detailed information on cards supported, and the most up to date information, refer to *Recommended Wireless Hardware*.

Cards Supporting Access Point (hostap) Mode

The cards in this section support acting as an access point to accept connections from other wireless clients. This is referred to as *hostap* mode.

ath(4)

The *ath(4)* driver supports cards based on the Atheros AR5210, AR5211, AR5212, AR5416, and AR92xx APIs which are used by many other Atheros chips of varying model numbers. Most Atheros cards support four virtual access points (VAPs) or stations or a combination to create a wireless repeater.

Though not explicitly listed in the man page, the FreeBSD Wiki Article for 802.11n Support also states that the driver has support for AR9130, AR9160, AR9280, AR9285, AR9287, and potentially other related chipsets.

ral(4) / ural(4) / rum(4) / run(4)

There are several related Ralink Technology IEEE 802.11 wireless network drivers, each for a different set and type of card.

- *ral(4)* supports cards based on the Ralink Technology RT2500, RT2501 and RT2600, RT2700, RT2900, and RT3090 chipsets.
- *ural(4)* supports RT2500USB.
- *run(4)* supports RT2700U, RT2800U, RT3000U, RT3900E, and similar.
- *rum(4)* supports RT2501USB and RT2601USB and similar.

Of these, only certain chips supported by *run(4)* can support VAPs.

The RT3090 *ral(4)* chip is the only model listed as capable of 802.11n on FreeBSD. The RT2700 and RT2800 *ral(4)* and the RT3900E *run(4)* hardware are capable of 802.11n, but the drivers on FreeBSD do not currently support their 802.11n features.

mwl(4)

The Marvell IEEE 802.11 wireless network driver, *mwl(4)*, supports cards based on the 88W8363 chipset and fully supports 802.11n. This card supports multiple VAPs and stations, up to eight of each.

Cards Only Supporting Client (station) Mode

The cards in this section are not capable of acting as access points, but may be used as clients in station mode, for example as a wireless WAN.

uath(4)

Atheros USB 2.0 wireless devices using AR5005UG and AR5005UX chipsets are supported by the *uath(4)* driver.

ipw(4) / iwi(4) / iwn(4) / wpi(4)

Intel wireless network drivers cover various models with different drivers.

- *ipw(4)* supports Intel PRO/Wireless 2100 MiniPCI adapters.
- *iwi(4)* supports Intel PRO/Wireless 2200BG/2915ABG MiniPCI and 2225BG PCI adapters.
- *iwn(4)* supports Intel Wireless WiFi Link 4965, 1000, 5000 and 6000 series PCI-Express adapters.
- *wpi(4)* supports Intel 3945ABG adapters.

Cards supported by the *iwn(4)* driver are documented by FreeBSD as supporting 802.11n in client mode.

Several Intel adapters have a license restriction with a warning that appears in the boot log. The *ipw(4)*, *iwi(4)*, and *wpi(4)* drivers have license files that must be read and agreed to. These license are located on the firewall in */usr/share/doc/legal/intel_ipw/LICENSE*, */usr/share/doc/legal/intel_iwi/LICENSE*, and */usr/share/doc/legal/intel_wpi/LICENSE* respectively. To agree to the license, edit */boot/loader.conf.local* and add a line to indicate the license acknowledgment, such as:

```
legal.intel_ipw.license_ack=1
```

Given the limited use of these adapters as clients only, a GUI-based solution to acknowledge these licenses has not yet been created.

bwi(4) / bwn(4)

The Broadcom BCM43xx IEEE 802.11b/g wireless driver is split in two depending on the specific models in use.

- *bwi(4)* supports BCM4301, BCM4303, BCM4306, BCM4309, BCM4311, BCM4318, BCM4319 using an older v3 version of the Broadom firmware.
- *bwn(4)* supports BCM4309, BCM4311, BCM4312, BCM4318, BCM4319 using a newer v4 version of the Broadcom firmware.

Support offered by the drivers does overlap for some cards. The *bwn(4)* driver is preferred for the cards it supports while the *bwi(4)* driver must be used on the older cards not covered by *bwn(4)*

malo(4)

Marvell Libertas IEEE 802.11b/g wireless driver, *malo(4)*, supports cards using the 88W8335 chipset.

upgt(4)

The Conexant/Intersil PrismGT SoftMAC USB IEEE 802.11b/g wireless driver, *upgt(4)*, supports cards using the GW3887 chipset.

urtw(4) / urtwn(4) / rsu(4)

The trio of related Realtek wireless drivers cover several different models:

- *urtw(4)* supports RTL8187B/L USB IEEE 802.11b/g models with a RTL8225 radio
- *urtwn(4)* supports RTL8188CU/RTL8188EU/RTL8192CU 802.11b/g/n
- *rsu(4)* supports RTL8188SU/RTL8192SU 802.11b/g/n

As in other similar cases, though the chips supported by *urtwn(4)* and *rsu(4)* are capable of 802.11n, FreeBSD does not support their 802.11n features.

zyd(4)

The ZyDAS ZD1211/ZD1211B USB IEEE 802.11b/g wireless network device driver, *zyd(4)*, supports adapters using the ZD1211 and ZD1211B USB chips.

31.3 Working with Virtual Access Point Wireless Interfaces

pfSense® software supports virtual wireless interfaces using Multi-BSS. These are known as Virtual Access Point or VAP interfaces, even if they are being used for client mode. VAPs allow multiple access points or clients to be run on the same wireless card, or to use a combination of access point and client mode. The most common use case is for multiple access points with different SSIDs each with unique security requirements. For example, one with no encryption but with captive portal and strict access rules and a separate network with encryption, authentication, and less strict access rules.

Even if a card does not support multiple VAP instances, the first entry must be created manually before it can be assigned.

Support for VAPs varies by card and driver, consult the information on driver support in *Recommended Wireless Hardware* to learn more. Odds are, however, if an Atheros wireless card is in use, it will work. While there is no theoretical limit to the number of VAPs a card may use, driver and hardware support varies, so the practical limit is four VAPs on *ath(4)* and eight on *mwl(4)*.

All VAPs on a given card share some common settings, such as the channel, regulatory settings, antenna settings, and wireless standard. Other settings such as the mode, SSID, encryption settings and so on may vary between VAPs.

31.3.1 Creating and Managing Wireless Instances

To create a new wireless instance:

- Navigate to **Interfaces > Assignments** on the **Wireless** tab.
- Click  **Add** to create a new entry
- Select the **Parent Interface**, for example *ath0*
- Pick the **Mode** from one of *Access Point*, *Infrastructure* (BSS, client mode), or *Ad-hoc* (IBSS)
- Enter a **Description**
- Click **Save**

An example is shown in Figure *Adding a Wireless Instance*.

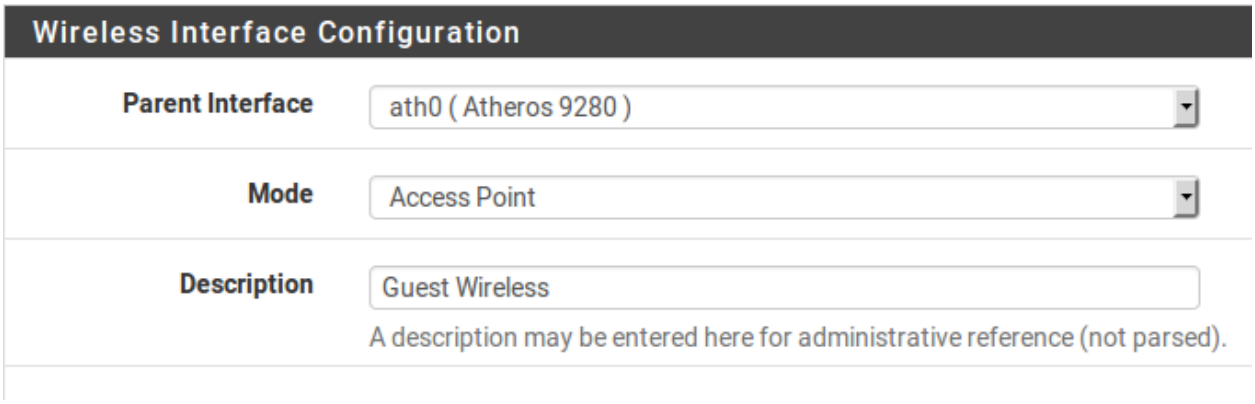


Fig. 1: Adding a Wireless Instance

Once the entry has been saved it is then available for assignment under **Interfaces > Assignments**. From there, assign and then edit the settings like any other wireless interface.

Note: The assigned interface must be configured to use the same mode specified when the VAP was created.

31.4 pfSense as an Access Point

With a wireless card that supports hostap mode (See *Cards Supporting Access Point (hostap) Mode*), pfSense® software can be configured as a wireless access point.

31.4.1 Should an external AP or pfSense be used for an access point?

The access point functionality in FreeBSD, and thus pfSense, has improved dramatically over the years and is considered stable currently for most uses. That said, many use cases behave better with an external access point, especially deployments that have requirements such as 802.11ac, concurrent operation in 2.4GHz and 5GHz, wireless mesh networks, or rare cases with clients that will not associate with an access point run using pfSense.

Access points on pfSense have been used with success in small-to-medium deployments, with gear such as a MacBook Pro, Apple AirTunes, iPod Touch, iPad, Android phones and tablets, various Windows laptops, Xbox, and FreeBSD clients and it works very reliably across all these devices. There is the possibility of finding incompatible devices with any access point, and FreeBSD is no exception.

The main deciding factor these days is 802.11n or 802.11ac support; Support for 802.11n hardware in pfSense is somewhat limited and 802.11ac support does not exist. This is a deal breaker for some, and as such using an external access point would be best for networks requiring 802.11ac and in some cases 802.11n if suitable hardware cannot be obtained.

The next most common factor is location of the antennas or the wireless access point in general. Often, the firewall running pfSense is located in an area of the building that is not optimal for wireless, such as a server room in a rack. For ideal coverage, the best practice is to locate the AP in an area that is less susceptible to wireless interference and that would have better signal strength to the area where wireless clients reside. If the firewall running pfSense is located alone on a shelf in a common area or other similar area conducive to good wireless signal, this may not be a concern.

31.4.2 Configuring pfSense as an access point

The process of configuring pfSense to act as a wireless access point (AP) is relatively easy. Many of the options will be familiar to anyone who has configured other wireless routers before, and some options may be new unless commercial-grade wireless equipment has been used. There are dozens of ways to configure access points, and they all depend upon the environment in which it will be deployed. In this example pfSense is configured as a basic AP that uses WPA2 encryption with AES. In this example, ExampleCo needs wireless access for some laptops in the conference room.

Preparing the Wireless Interface

Before starting, ensure that the wireless card is installed in the firewall and the pigtails and antennas are firmly attached.

Create the wireless instance as described in *Creating and Managing Wireless Instances* if it does not already exist. When working as an access point, it must use *Access Point* mode. The wireless card must be assigned as an OPT interface and enabled before the remaining configuration can be completed.

Interface Description When in use as an access point, naming the interface *WLAN* (Wireless LAN) or *Wireless*, or naming it after the SSID makes it easier to identify. If pfSense will be driving multiple access points, there should be some way to distinguish them, such as "WLANadmin" and "WLANsales". In this example, it is named `ConfRoom`.

Interface Type Since this example will be an AP on a dedicated IP subnet, the IPv4 Configuration Type must be set to *Static IPv4*

IP Address An IPv4 Address and subnet mask must be specified. This is a separate subnet from the other interfaces. For this example it can be `192.168.201.0/24`, a subnet that is otherwise unused in the ExampleCo network. Using that subnet, the IPv4 Address for this interface will be `192.168.201.1`.

Common Wireless Settings

These settings are shared for all VAPs on a given physical wireless card. Changing these settings on one interface will change them on all other virtual interfaces using the same physical adapter.

Persist common settings By checking **Persist common settings**, the configuration values in this section will be preserved even if all the interfaces and VAPs are deleted or reassigned, when they would otherwise be lost.

Wireless Standard Depending upon hardware support, there are several choices available for the wireless **Standard** setting, including *802.11b*, *802.11g*, *802.11g turbo*, *802.11a*, *802.11a turbo*, *802.11ng*, *802.11na*, and possibly others. For this example, we will choose *802.11ng* for an 802.11n access point operating in the 2.4GHz band.

802.11g OFDM Protection Mode The **802.11g OFDM Protection Mode** setting is only useful in mixed standard environments where 802.11g and 802.11b have to interact. Its primary use is for avoiding collisions. Given the age of 802.11b and scarcity of working devices that use it, the setting is best left at *Protection mode off*. There is a performance penalty for using it, since it has some overhead on each frame and also requires extra steps when transmitting frames.

Wireless Channel Selection When selecting a **Channel**, knowledge of nearby radio transmitters in similar frequency bands is required to avoid interference. In addition to wireless access points, there are also cordless phones, Bluetooth, baby monitors, video transmitters, microwaves, and many other devices that utilize the same 2.4 GHz spectrum that can cause interference.

Often any channel will work so long as the AP clients are near the antenna. With 802.11g and before, the safest channels to use were *1* , *6* , and *11* since their frequency bands did not overlap each other. This is no longer true with 802.11n and later or even some 802.11g setups which use wider ranges of frequencies to attain higher speeds. For this network, since there are no others around, channel *1* is a fine choice.

Note: Always pick a specific channel. Do not select *Auto* for the channel of an Access Point. The input validation on current versions of pfSense prevents this from being selected.

When using other standards, or using wireless in countries other than the US, there may be many more channels available than described here. Cards that support 802.11a or 802.11n may also support channels in the 5 GHz spectrum.

The full list of channels supported by the card is shown in the **Channel** drop- down and must agree with the chosen *Standard*. For example, do not choose *802.11ng* for the **Standard** and then pick a **Channel** used only for *802.11na*. The channel list also includes some information about the standard, frequency of the channel, and the maximum transmit power both of the card and in the regulatory domain for that particular channel. Be careful to watch the power when selecting a channel, because some channels, especially in the 5GHz band, vary widely in their allowed power levels.

See also:

Survey tools such as NetSurveyor, InSSIDer, Wi-Spy, and countless other apps for various operating systems, phones, tablets, and so on may help to choose a less busy channel or area of the spectrum. Mileage may vary.

Distance setting Measured in meters, and only supported by Atheros cards, The **Distance Setting** field tunes ACK/CTS timers to fit the distance between AP and Client. In most cases it is not necessary to configure this value, but it may help in certain tricky wireless setups such as long-range clients.

Regulatory settings

The **Regulatory settings** section controls how the card is allowed to transmit legally in a specified region. Different countries typically have different regulatory settings, and some countries have none. If unsure, check with the local government to see which laws apply in a given area. The default values are usually OK, as the cards may be set to a specific region already. In some cases **Regulatory settings** must be set manually if the card has a default not understood by the driver. Similar to the previous section, these values are applied to the card itself and cannot vary between VAPs on the card.

While it may be tempting to set the card to *Debug* in order to use settings not otherwise allowed, this action could result in legal trouble should it be noticed. The likelihood of this happening varies greatly by country/area so use that with caution.

Regulatory domain The **Regulatory Domain** is the governmental body that controls wireless communications in a region. For example, the US and Canada follow FCC regulations while in the UK it's ETSI. If unsure of the regulatory domain in a region, see the **Country** setting.

Country Sometimes specific countries inside a regulatory domain have different restrictions. The **Country** option contains a drop-down list of many countries throughout the world and their associated country codes and regulatory domains.

Location Certain restrictions exist for *Indoor* and *Outdoor* transmissions as well. Setting the **Location** of the transmitter will further adjust the allowed transmission power and/or channels.

Network-specific wireless configuration

These settings are unique per interface, even on virtual wireless interfaces. Changing these settings does not affect any other interfaces.

Wireless Mode Set the **Mode** field to *Access Point* , and pfSense will use `hostapd` to act as an AP.

Service Set Identifier (SSID) The **SSID** is the "name" of the AP as seen by clients. Set the SSID to something readily identifiable yet unique. Keeping with the example, *ConfRoom* is a good name to use.

Minimum wireless standard The **Minimum wireless standard** drop-down controls whether or not older clients are able to associate with this access point. Allowing older clients may be necessary in some environments if devices are still around that require it. Some devices are only compatible with 802.11g and require a mixed network g/n in order to work. The flip side of this is that slower speeds may be seen as a result of allowing such devices on the network as the access point will be forced to cater to the lowest common denominator when an 802.11g device is transmitting at the same time as an 802.11n device. In our example conference room, users will only be using recently purchased company-owned laptops that are all capable of 802.11n, so *802.11n* is the best choice.

Intra-BSS Communication If **Allow intra-BSS communication** is checked, wireless clients will be able to see each other directly. If clients will only need access to the Internet, it is typically safer to uncheck this option. In this scenario, users in the conference room may need to share files back and forth directly between laptops, so this will stay checked.

Enable WME Wireless Multimedia Extensions, or **WME**, is a part of the wireless standard that provides some Quality of Service for wireless traffic to ensure proper delivery of multimedia content. It is required for 802.11n to operate, but is optional for older standards. This feature is not supported by all cards/drivers.

Hide SSID Normally the AP will broadcast its SSID so that clients can locate and associate with it easily. This is considered by some to be a security risk, announcing to all who are listening that a wireless network is available, but in most cases the convenience outweighs the (negligible) security risk. The benefits of disabling SSID broadcasting are overblown by some, as it does not actually hide the

network from anyone capable of using many freely available wireless security tools that easily find such wireless networks. For our conference room AP, we will leave this unchecked to make it easier for meeting attendees to find and use the service.

Wireless Encryption (WPA)

Two types of encryption are supported for 802.11 networks: WPA, and WPA2. WPA2 with AES is the most secure. Even when not worrying about encrypting the over-the-air traffic (which should be done), it provides an additional means of access control. All modern wireless cards and drivers support WPA2.

Warning: Wireless Encryption Weaknesses

WEP has serious known security problems for years, and support for WEP has been removed from pfSense. It is possible to crack WEP in a matter of minutes at most, and it should never be relied upon for security. If WEP is required, an external AP must be used.

TKIP (Temporal Key Integrity Protocol), part of AES, became a replacement for WEP after it was broken. It uses the same underlying mechanism as WEP, and hence is vulnerable to some similar attacks. These attacks have become more practical and TKIP is no longer considered secure. TKIP should never be used unless devices are present that are incompatible with WPA or WPA2 using AES. WPA and WPA2 in combination with AES are not subject to these flaws in TKIP.

In this example, the ConfRoom wireless must be secured with WPA2.

Enable This checkbox enables *WPA* or WPA2 encryption, so it should be **checked**

WPA Pre-Shared Key Enter the desired wireless key, in this example `excoconf213`.

WPA Mode WPA or WPA2, in this example, *WPA2*

WPA Key Management Mode Can be *Pre-Shared Key* (PSK) or *Extensible Authentication Protocol* (EAP). In this example, PSK is sufficient.

WPA Pairwise This should almost always be set to *AES*, due to the weaknesses in *TKIP* mentioned previously.

Group Key Rotation This option allows setting how often the broadcast/multicast encryption keys (Group Transient Key, GTK) are rotated, in seconds. It can be any value from `1` to `9999` but it should be shorter than the **Group Master Key Regeneration** value. The default value of `60` seconds (one minute) is adequate. Lower values may be more secure but may bog things down with frequent rekeying.

Group Master Key Regeneration This parameter controls how often, in seconds, the master key (Group Master Key, GMK) used internally to generate GTKs is regenerated. It can be any value from `1` to `9999` but it should be longer than the **Group Key Rotation** value. The default value of `3600` seconds (one hour) is adequate.

Strict Key Regeneration This option causes the firewall to change the GTK whenever a client leaves the access point, much like changing the passwords when an employee leaves. There may be a slight performance penalty in cases where there is a high turnover of clients. In cases where security is not a primary concern, this can be left disabled.

IEEE 802.1X Authentication (WPA Enterprise)

Another type of supported wireless security is known as IEEE 802.1X Authentication, or more commonly referred to as *WPA Enterprise* or *WPA2 Enterprise*. This mode allows using a more traditional username and password entry in order to gain access to the wireless network. The downside is that this authentication must be done via RADIUS servers. If an existing RADIUS server is already present or easily deployed, it may be a viable source of wireless access control. In this example, 802.1X is not used but the options are explained.

See also:

The FreeRADIUS package (*FreeRADIUS package*) can fulfill this purpose.

Note: Some older operating systems may not properly handle 802.1X or may have long delays after failed authentication attempts, but there are typically workarounds for those issues via OS updates or patches.

Clients must also be configured to properly access the service. Some may pick up the proper settings automatically, others may need set for a specific mode (e.g. *PEAP*) or may need certificates loaded. The specific values depend on the RADIUS server settings.

To get started with 802.1X authentication, first set **WPA Key Management** to *Extensible Authentication Protocol*.

Enable 802.1X Authentication When checked, 802.1X authentication support is enabled and required of clients.

Primary 802.1X Server The preferred server for 802.1X authentication.

IP Address The IP address of the preferred RADIUS server to use for 802.1X client authentication.

Port The port upon which to contact the RADIUS server for authentication requests, typically `1812`.

Shared Secret The password to use when communicating with the RADIUS server from this firewall. This must match the shared secret defined for this firewall on the RADIUS server.

Secondary 802.1X Server The same parameters as above, but for a secondary RADIUS server in case the first one is unreachable.

Authentication Roaming Preauth This option sets up pre-authentication to speed up roaming between access points. This will perform part of the authentication process before the client fully associates to ease the transition.

Finishing AP Settings

The previous settings are enough to get a wireless access point running with 802.11n with WPA2 + AES encryption. When the settings are complete, click **Save**, then **Apply Changes**.

Configuring DHCP

Now that an entirely separate network has been created, DHCP must be enabled to automatically provide associating wireless clients an IP address. Browse to **Services > DHCP Server**, click on the tab for the wireless interface (**ConfRoom** for this example). Check the box to **Enable**, set whatever size range will be needed, and any additional options desired, then click **Save** and **Apply Changes**. For more details on configuring the DHCP service, see *DHCP*.

Adding Firewall Rules

Since this wireless interface is an OPT interface, it will have no default firewall rules. At the very least a rule must be added to allow traffic from this subnet to any destination. Since the conference room users will need internet access and access to other network resources, a default allow rule will be fine in this case. To create the rule:

- Navigate to **Firewall > Rules**
- Click on the tab for the wireless interface (**ConfRoom** for this example).
- Click  **Add** and configure a rule as follows:

  Interface *ConfRoom*

  Protocol *Any*

  Source *ConfRoom Net*

  Destination *Any*
- Click **Save**
- Click **Apply Changes**

See also:

For more information about creating firewall rules, see *Firewall*.

Associating Clients

The newly configured pfSense AP should appear in the list of available access points from a wireless device, assuming broadcasting of the SSID was not disabled. A client should now be able to associate with it as it would with any other access point. The exact procedure will vary between operating systems, devices, and drivers, but most manufacturers have streamlined the process to make it simple for everyone.

Viewing Wireless Client Status

When a wireless interface is configured for access point mode, the associated clients will be listed on **Status > Wireless**.

Interesting sysctls from shell that cannot be controlled from GUI

dev.ath.0.tpc 0 = disable 1 = enable Switch on or off Transmission Power Control. Can be tricky in point to multipoint applications.

dev.ath.0.tpscale 0,1,2,3,4 Size of the increment that TPC will use to up/down the power, normally 1 is the best choice. A higher scale value will most likely make the link drop if the signal is close to what it needs to be and the TPC is throttled down.

dev.ath.0.tpack 0 -> 99 Controls the ACK power separately. Normally it is the same as tpcts

dev.ath.0.tpcts 0 -> 99 Controls the CTS power separately. Normally it is the same as tpack

Tuning ACK timers manually:

```
Real life values:

range                  ack-timeout

5GHz  5GHz-turbo       2.4GHz-G
0km   default default          default
5km   52      30               62
10km  85      48               96
15km  121     67               133
20km  160     89               174
25km  203     111              219
30km  249     137              268
35km  298     168              320
40km  350     190              375
45km  405     -                -
```

31.5 Wireless WAN

A wireless card in a firewall running pfSense® software can be used as the primary WAN interface or an additional WAN in a multi-WAN deployment.

31.5.1 Interface assignment

If the wireless interface has not yet been assigned, there are two possible choices: Add it as an additional OPT interface or reassign it as WAN.

Before starting, create the wireless instance as described in *Creating and Managing Wireless Instances* if it does not already exist. When working as a WAN, it must use *Infrastructure* mode (BSS).

To add the interface as a new OPT interface:

- Browse to **Interfaces > Assignments**
- Select the wireless interface from the **Available network ports** drop-down below the other interfaces
- Click  **Add** to add the interface as an OPT interface

To reassign the wireless interface as WAN:

- Browse to **Interfaces > Assignments**
- Select the wireless interface as **WAN**

- Click Save

Figure *Wireless WAN Interface Assignment* shows an Atheros card assigned as WAN.

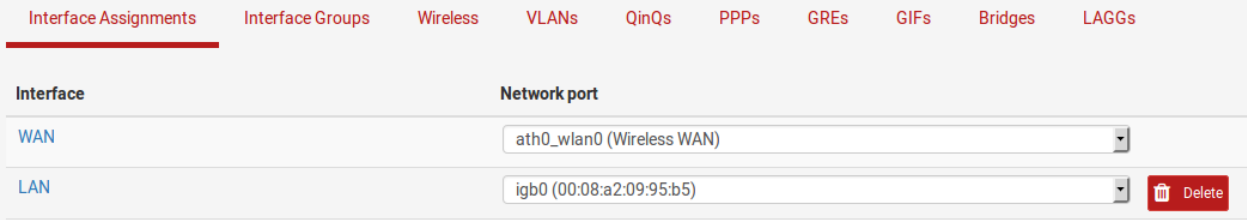


Fig. 2: Wireless WAN Interface Assignment

31.5.2 Configuring the wireless network

Most wireless WANs need only a handful of options set, but specifics vary depending on the Access Point (AP) to which this client interface will connect.

- Browse to the **Interfaces** menu for the wireless WAN interface, for example **Interfaces > WAN**
- Select the type of configuration (*DHCP*, *Static IP*, etc.)
- Scroll down to **Common Wireless Configuration**
- Set the **Standard** to match the AP, for example *802.11g*
- Select the appropriate **Channel** to match the AP
- Scroll down to **Network-specific Wireless Configuration**
- Set the **Mode** to *Infrastructure (BSS)* mode
- Enter the **SSID** for the AP
- Configure encryption such as WPA2 (Wi-Fi Protected Access) if in use by the AP
- Review the remaining settings if necessary and select any other appropriate options to match the AP
- Click **Save**
- Click **Apply Changes**

31.5.3 Checking wireless status

Browse to **Status > Interfaces** to see the status of the wireless interface. If the interface has successfully associated with the AP it will be indicated on the status page. A **status** of **associated** means the interface has connected to the AP successfully, as shown in Figure *Associated Wireless WAN Interface*

If the interface **status** shows **No carrier**, it was unable to associate. Figure *No carrier on wireless WAN* shows an example of this, where the antenna was disconnected so it could not connect to a wireless network that was some distance away.

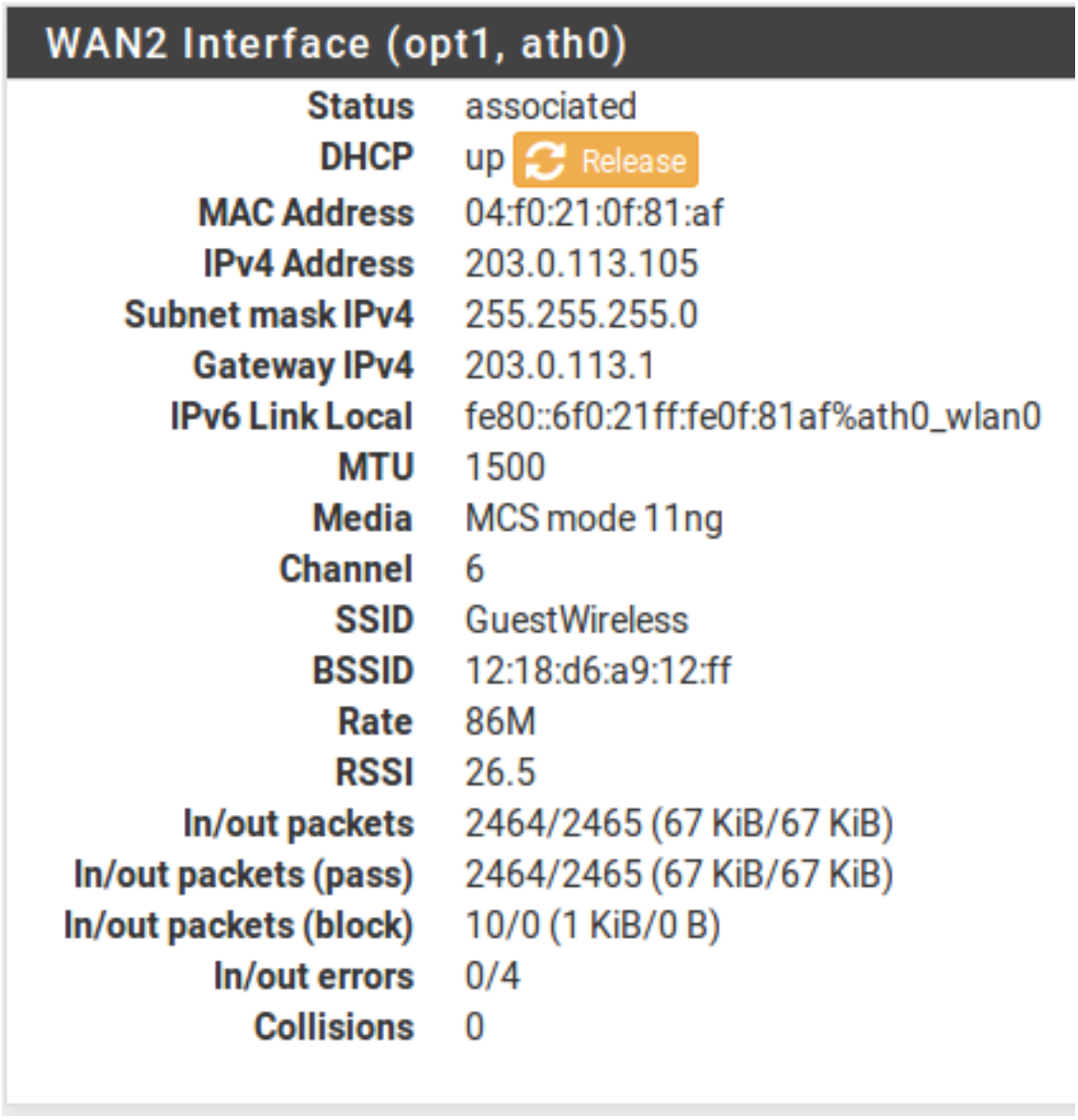


Fig. 3: Associated Wireless WAN Interface

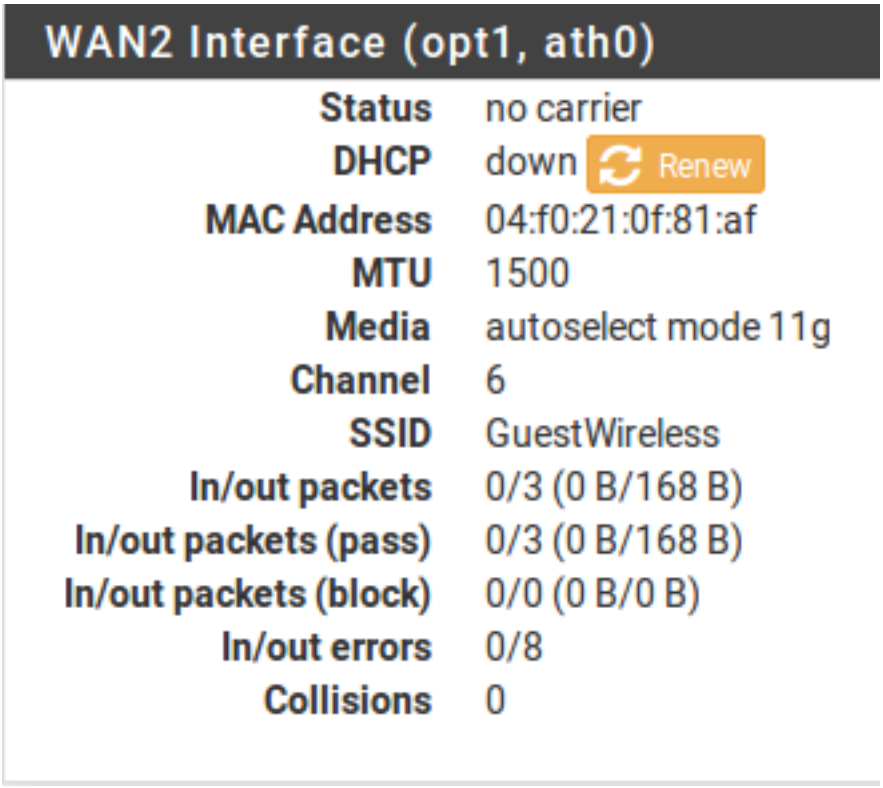


Fig. 4: No carrier on wireless WAN

31.5.4 Showing available wireless networks and signal strength

The wireless access points visible by the firewall may be viewed by navigating to **Status > Wireless** as shown in Figure *Wireless Status*.

A wireless interface must be configured before this menu item will appear.

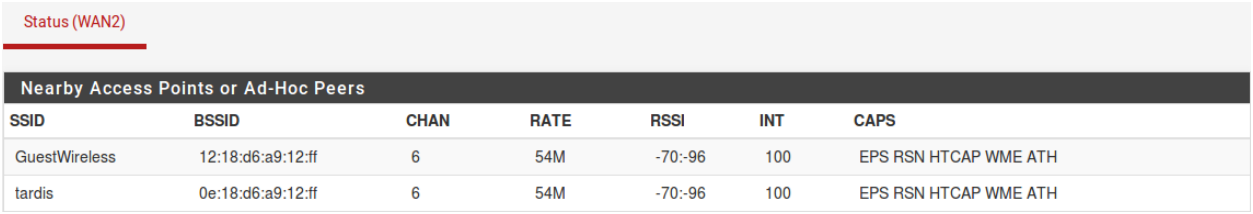


Fig. 5: Wireless Status

31.6 Bridging and wireless

Bridging two interfaces together places them on the same broadcast domain as if they were connected to the same switch. Typically this is done so that two interfaces will act as though they are on the same flat network using the same IP subnet, in this case a wireless interface and wired interface. When two interfaces are bridged, broadcast and multicast traffic is forwarded to all bridge members.

Certain applications and devices rely on broadcast traffic to function. For example, Apple's AirTunes will not function across two broadcast domains. So if AirTunes is present on the wireless network and it must be accessed from a system on the wired network, the wired and wireless networks must be bridged. Other examples include media services provided by devices such as Chromecast, TiVo, Xbox 360, and Playstation 3. These rely on multicast or broadcast traffic that can only function if the wired and wireless networks are bridged.

31.6.1 Wireless Access Points and Bridging

Only wireless interfaces in access point (hostap) mode will function in a bridged configuration. A wireless interface configured for hostap can be bridged to another interface which combines them on the same broadcast domain. This may be desirable for certain devices or applications that must reside on the same broadcast domain to function properly, as mentioned previously.

31.6.2 BSS and IBSS wireless and Bridging

Due to the way wireless works in BSS mode (Basic Service Set, client mode) and IBSS mode (Independent Basic Service Set, Ad-Hoc mode), and the way bridging works, a wireless interface cannot be bridged in BSS or IBSS mode. Every device connected to a wireless card in BSS or IBSS mode must present the same MAC address. With bridging, the MAC address passed is the actual MAC of the connected device. This is normally a desirable facet of how bridging works. With wireless, the only way this can function is if all the devices behind that wireless card present the same MAC address on the wireless network. This is explained in depth by noted wireless expert Jim Thompson in a mailing list post.

As one example, when VMware Player, Workstation, or Server is configured to bridge to a wireless interface, it automatically translates the MAC address to that of the wireless card. Because there is no way to translate a MAC address in FreeBSD, and because of the way bridging in FreeBSD works, it is difficult to provide any workarounds similar to what VMware offers. At some point pfSense® may support this, but it is not on the road map for 2.x.

31.6.3 Choosing Routing or Bridging

The choice between bridging (using the same IP subnet as the existing LAN) or routing (using a dedicated IP subnet for wireless) for wireless clients will depend on what services wireless clients require. In many home network environments there are applications or devices that require wired and wireless networks to be bridged. In most corporate networks, there are few if any applications that require bridging. Which to choose depends on the requirements of network applications in use, as well as personal preference.

There are some compromises to this, one example being the Avahi package. It can listen on two different broadcast domains and rebroadcast messages from one to the other in order to allow multicast DNS to work (aka Rendezvous or Bonjour) for network discovery and services. If the required services all use protocols that can be handled by Avahi, then using a routed method may be possible.

For services running on the firewall, bridging can also be problematic. Features such as limiters, Captive Portal, and transparent proxies require special configuration and handling to work on bridged networks. Specifically, the bridge itself must be assigned and the only interface on the bridge with an IP address must be the assigned bridge. Also, in order for these functions to work, the IP address on the bridge must be the address used by clients as their gateway.

31.7 Additional protection for a wireless network

In addition to strong encryption from WPA2 with AES, some users like to employ an additional layer of encryption and authentication before allowing access to network resources. The two most commonly deployed solutions are *Captive Portal* and *Virtual Private Networks*. These methods can be employed whether an external access point is used on an OPT interface or an internal wireless card as the access point.

Note: In theory, The PPPoE server could also be used in this role but support would be impossible on some clients and non-trivial on most others, so it is not typically a viable option when combined with wireless.

31.7.1 Additional wireless protection with Captive Portal

By enabling Captive Portal on the interface where the wireless resides, authentication can be required before users may access network resources beyond the firewall. In corporate networks, this is commonly deployed with RADIUS authentication to Microsoft Active Directory so users can use their Active Directory credentials to authenticate while on the wireless network. Captive Portal configuration is covered in *Captive Portal*.

Note: If the sole requirement is per-user RADIUS authentication, a better solution for RADIUS is 802.1X rather than using Captive Portal, unless there are clients present which do not support 802.1X.

Captive Portal is more likely to be used on open or limited access wireless networks, such as those in a hotel, restaurant, or similar setting where there is either no encryption enabled or a common knowledge/shared key.

31.7.2 Additional protection with VPN

Adding Captive Portal provides another layer of authentication, but does not offer any additional protection from eavesdropping of wireless traffic. Requiring a VPN connection before allowing access to the internal network and Internet adds another layer of authentication as well as an additional layer of encryption for wireless traffic. The configuration for the chosen type of VPN will be no different from a remote access configuration, but the firewall rules must be configured on the pfSense® interface to only allow VPN traffic from wireless clients.

Configuring firewall rules for IPsec

Figure *Rules to allow only IPsec from wireless* shows the minimal rules required to allow only access to IPsec on the WLAN interface IP address. Pings to the WLAN interface IP address are also allowed to assist in troubleshooting.

Configuring firewall rules for OpenVPN

Figure *Rules to allow only OpenVPN from wireless* shows the minimal rules required to allow access only to OpenVPN on the WLAN interface IP address. Pings to the WLAN interface IP address are also allowed to assist in troubleshooting. This assumes the default UDP port 1194 is in use. If another protocol or port was chosen, adjust the rule accordingly.

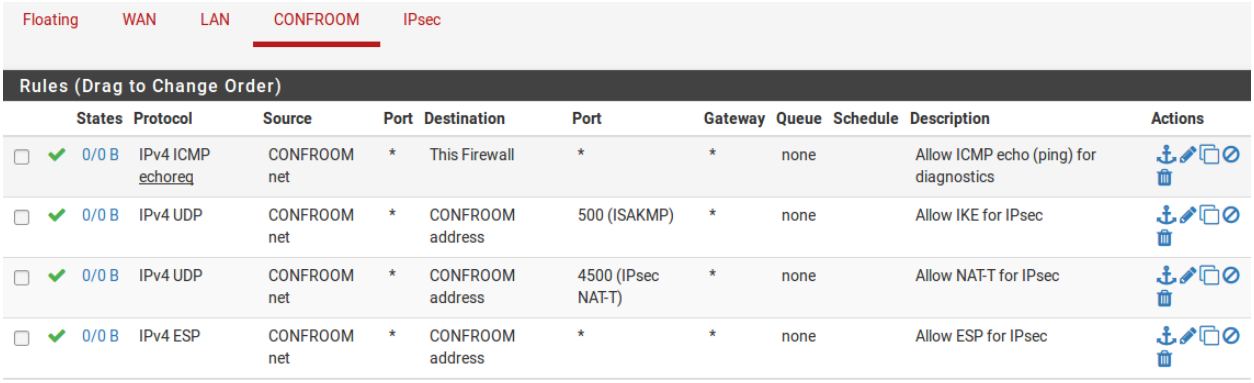


Fig. 6: Rules to allow only IPsec from wireless

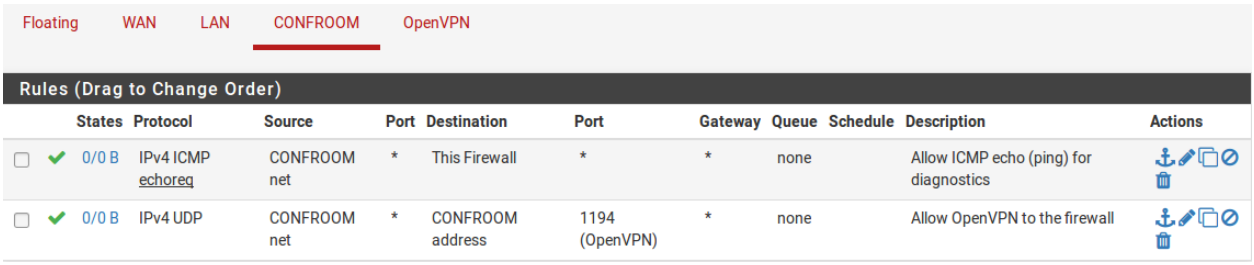


Fig. 7: Rules to allow only OpenVPN from wireless

31.8 Configuring a Secure Wireless Hotspot

A company or organization may wish to provide Internet access for customers or guests using an existing Internet connection. This can be a boon to the customers and business, but can also expose the existing private network to attack if not done properly. This section covers the common means of providing Internet access to guests and customers, while protecting the internal network.

31.8.1 Multiple firewall approach

For the best protection between a private network and a public network, obtain at least two public IP addresses from the ISP and use a second firewall for the public network. To accommodate this, place a switch between the Internet connection and the WAN interface of both firewalls.

This has the added benefit of putting the public network on a different public IP address from the private network, so if a report of abuse is received, it is easy to tell the origin. The firewall protecting the private network will see the public network no differently than any Internet host and vice versa.

31.8.2 Single firewall approach

In environments where the multiple firewall approach is cost prohibitive or otherwise undesirable, the internal network can still be protected by connecting the public network to an OPT interface on a firewall running pfSense®. Assign a dedicated private IP subnet to this OPT interface, and configure its firewall rules to allow access to the Internet but not the internal network.

In *Rules to allow only Internet access from wireless* the firewall rules allow clients to reach the firewall for DNS requests and ICMP echo request (ping), but prevent all access to other private networks. The RFC1918 alias referenced in the figure includes the RFC1918 private network list, `192.168.0.0/16`, `172.16.0.0/12`, and `10.0.0.0/8`.

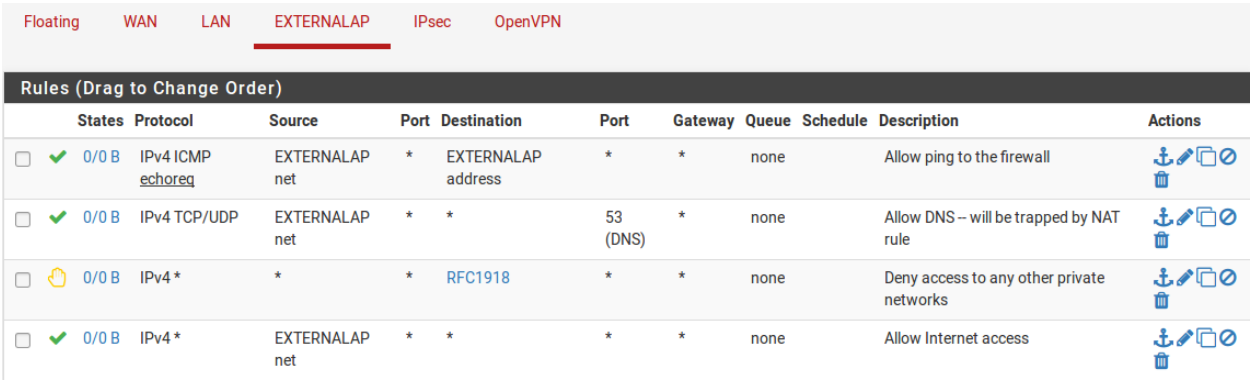


Fig. 8: Rules to allow only Internet access from wireless

31.8.3 Access control and egress filtering considerations

Other than not allowing traffic from the publicly accessible network to the private network, there are additional things to consider in the configuration of a hotspot.

Restrict network access

While many hotspots use open wireless networks with no other authentication, consider additional protections to prevent network abuse. On wireless, consider using WPA or WPA2 and providing the passphrase to guests or customers. Some businesses taking this approach display the passphrase on a placard in the lobby or waiting area, posted in a guest room, or provide it upon request. Also consider implementing Captive Portal on pfSense (covered in *Captive Portal*). This helps prevent people in other offices and outside the building from using the wireless network even if it is open access.

Disable Intra-BSS communication

If the access point allows, disable intra-BSS communication. This option is also sometimes called "AP Client Isolation". This prevents wireless clients from communicating with other wireless clients directly, which protects users from intentional attacks from other wireless users as well as unintentional ones such as worms.

Intra-BSS communication may be required for certain functions such as wireless printers, Chromecast devices or similar cases when two wireless devices must talk directly to each other, but this is rarely if ever required in the context of a public hotspot.

Egress filtering

Consider what kind of egress policy to configure. The most basic, allowing access to the Internet without allowing access to the private network, is probably the most commonly deployed but consider additional restrictions.

To avoid having the public IP address of the firewall black listed because of infected visiting systems acting as spam bots, consider blocking SMTP. Several large ISPs already block SMTP outbound because clients have moved to using authenticated access on the submission port (587) rather than using port 25 directly. An alternative that still lets people use their SMTP e-mail but limits the effect of spam bots is to create an allow rule for SMTP and specify **Maximum state entries per host** under **Advanced Options** when editing the firewall rule. Ensure the rule is above any other rules that would match SMTP traffic, and specify a low limit. Because connections may not always be properly closed by the mail client or server, do not set this too low to prevent blocking legitimate users, but a limit of five connections should be reasonable. **Maximum state entries per host** can be set on all firewall rules, but keep in mind that some protocols will require dozens or hundreds of connections to function. HTTP and HTTPS may require numerous connections to load a single web page depending on the content of the page and the behavior of the browser, so don't set the limits too low.

Balance the desires of users against the risks inherent in providing Internet access for uncontrolled systems, and define a policy that fits the environment.

See also:

- *Wireless Logs*
- *Wireless Status*
- *Using an External Wireless Access Point*
- *Troubleshooting Wireless Connections*

pfSense® software includes built in wireless capabilities that allow a firewall running pfSense software to be turned into a wireless access point, to use a wireless 802.11 connection as a WAN connection, or both. This chapter covers how to configure pfSense for these roles as well as suggested means of securely accommodating external wireless access points and how to securely deploy a wireless hotspot. In-depth coverage of 802.11 in general is outside the scope of this documentation. For those seeking such information, see other works such as 802.11 Wireless Networks: The Definitive Guide.

See also:

pfSense Hangouts on Youtube to view the May 2015 Hangout on Wireless Access Points.

CHAPTER

THIRTYTWO

CELLULAR WIRELESS

pfSense® software can use a supported cellular modem (3G/4G/LTE) as a WAN interface for connectivity. This can be used as a sole means of connectivity or as a backup.

32.1 Configuring 3G modems

To configure a 3G modem in pfSense® software on a current supported release, plug in a *Known Working Modem* and log into the web interface to begin configuration.

32.1.1 Configuring PPP

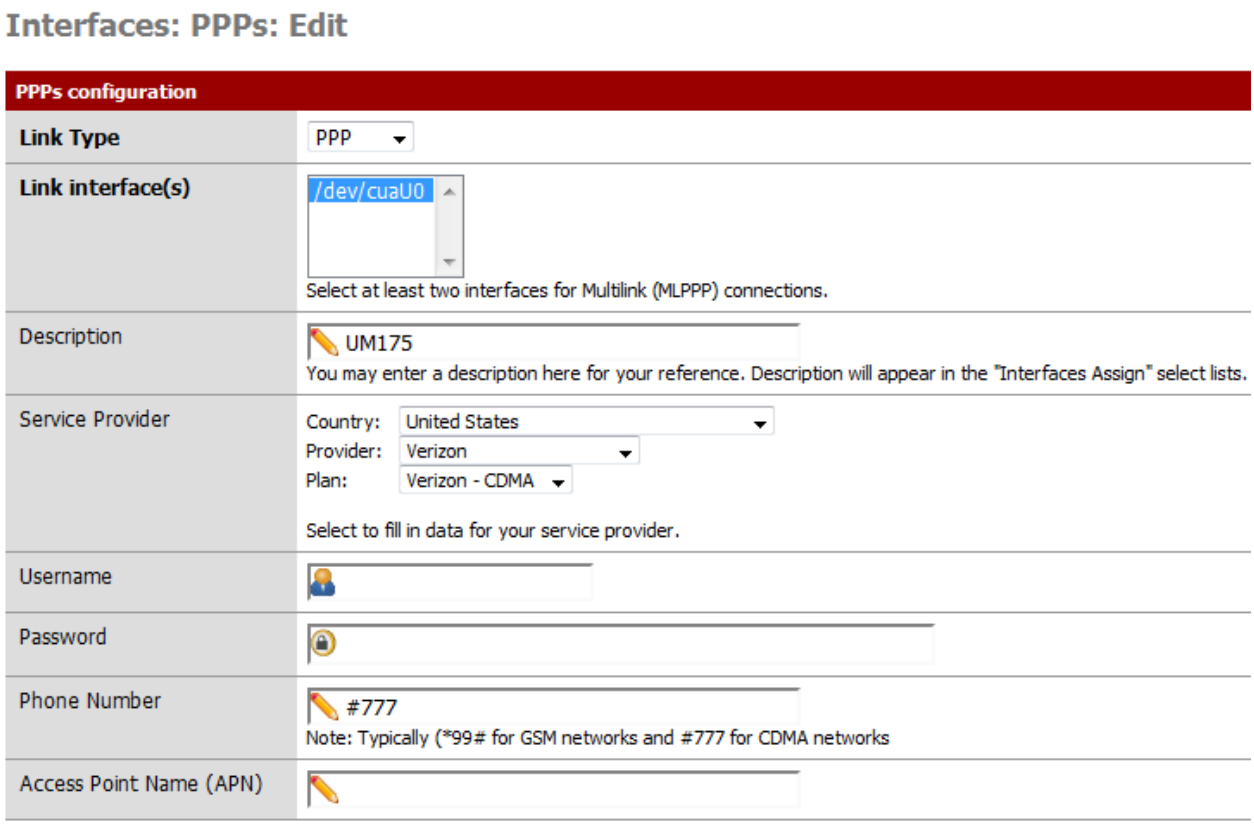
Browse to **Interfaces > Assignments**, and click the **PPPs** tab. Click **+** on that screen.

In the **Link Type** drop down, select *PPP*.

In the **Link interface(s)** box, the list will be populated with serial ports on the system. Select the port for the modem. A modem may list several serial ports. Typically it is the last one, but may require some trial and error. Future versions of pfSense software may properly auto-detect modems but that has historically been a source of problems.

Optionally fill in a **Description**, which will be used to reference this PPP configuration in other parts of the web interface.

Under **Service Provider**, select the **Country**. The **Provider** list for that country will appear, then select the provider of the card. Then in the **Plan** drop down, select the plan. This should adequately fill in all the PPP details needed for the connection. Click **Save**.

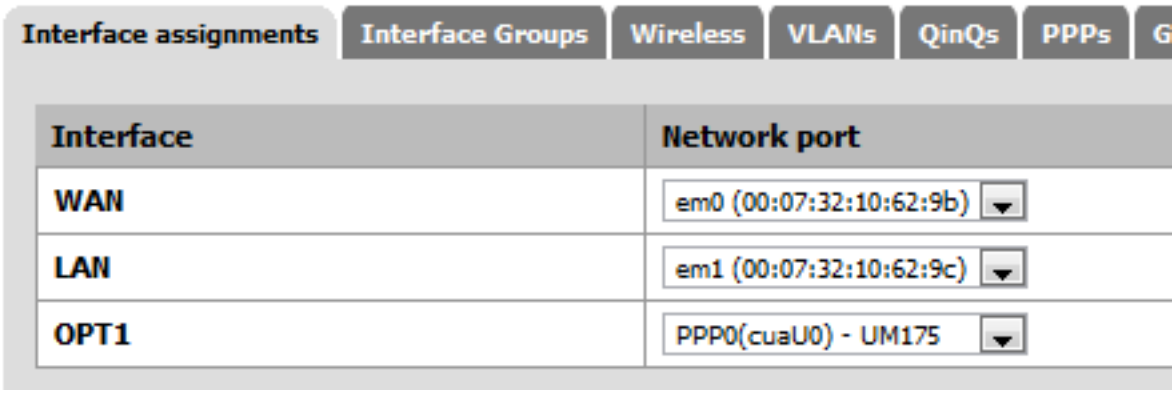


At the PPPs screen, the newly created PPP interface will be listed.



32.1.2 Assigning the PPP Interface

Next the PPP interface must be created. Click the **Interface assignments** tab, and click the **+** to add a new interface. Select the PPP interface, click **Save**, then **Apply changes**.



32.1.3 Enable the PPP Interface

Now browse to **Interfaces > OPT1** (or the interface name shown for the the PPP interface when it was assigned above). Check the **Enable Interface** box, rename it if desired, and click **Save**. Do not change anything else on the page. Then click **Apply changes**.

32.1.4 Check the interface status

Browse to **Status > Interfaces** to check the status of the newly created and assigned PPP interface. If it does not show connected, check the logs under **Status > System logs**, **PPP** tab to see why its connection is failing.

Note that some connection problems are a lack of signal. If the 3G modem is in a location with poor reception, such as an equipment room, datacenter, rack, etc, then it may be advisable to use a longer USB cable and/or Antenna to achieve a better signal.

32.2 Known Working 3G-4G Modems

This page lists 3G and 4G modem devices which are known to work with pfSense® software.

- 4G Systems XS Stick P14
- Alcatel Onetouch 4G L850V
- Anydata ADU-635 WA
- Verizon/Pantech UM175
- Verizon/Pantech UML290 and UML295 (see *here for UML290 config info*)
- Verizon USB727
- HP hs2340 HSPA+ Mini Card AMO Ericsson
- Huawei B970/B970B
- Huawei E122
- Huawei E153
- Huawei E156G
- Huawei E160
- Huawei E161
- Huawei E160E
- Huawei E169
- Huawei E172
- Huawei E173
  - It has been reported that some E173 modems are shipping labeled as E173 but have a different chip that is not supported. See below.
- Huawei E173U-1
- Huawei E176
- Huawei E180
- Huawei E220

- Huawei E226
- Huawei E272
- Huawei E352
- Huawei E353U-2
- Huawei E367
- Huawei E372
- Huawei E392
- Huawei E397u-53
  - Link interface: `/dev/cuaU0.0`
  - Init string: `&F`
- Huawei E398 (E398u-1)
- Huawei E960
- Huawei E1550
- Huawei E1552
- Huawei E1556
- Huawei E1612
- Huawei E1692
- Huawei E1750
- Huawei E1756
- Huawei E1762
- Huawei E1820
- Huawei E3372s LTE USB-stick
  - Link interface: `/dev/cuaU0.1`
  - Init string: `&F&C1&D2E0S0=0`
- Huawei E3372h LTE USB-stick
  - As an Ethernet device: see *Modems reported to work as Ethernet devices*
  - As a modem device: requires manual firmware changes, see this article
- Huawei E3531 - Command to switch to the correct mode:

```
usb_modeswitch -v 12d1 -p 15e7 -c /usr/local/share/usb_modeswitch/12d1:15e7
```

  The command has to be executed every time it's detached and reattached, the interface has to be disabled and then enabled again.
  - Link interface: `/dev/cuaU0.0`
  - Init string: `&F0E1Q0 +CMEE=2`
- Huawei K3563
- Huawei E5372
- Huawei E5776

- Huawei VIK K3715 HSU by Vodafone
- Huawei K3765 by Vodafone
- Huawei ME909u-521 4G/LTE Mini-PCIe
  - Link interface: `/dev/cuaU0.0`
  - Init string: `&F0E1Q0 +CMEE=2`
- Huawei ME909s-120 4G/LTE Mini-PCIe
- Nokia Phone E72-1 connected via USB cable
- Sierra Wireless U305
- Sierra Wireless 320u usb LTE
- Sierra Wireless U330
- Sierra Wireless MC7354
- Sierra Wireless MC7355
- Sierra Wireless MC7710
  - May require usb-comp or ID change. See https://forum.netgate.com/post/556751 for more information.
- HP2300 (Sierra Wireless MC8775 3G) Mini-PCIe
- USB Connect Mercury (Sierra Wireless Compass 885 or C885)
- Sierra Wireless Compass 889
- Ovation U727 by Novatel on Sprint CDMA
- Nokia CS-17
- Turkey-TTNET Usb Stick 3G Modem. Label says Huawei E173 but its actually Huawei E1800.
- Telstra maxon bp3-usb (Benchmarked: 2500/350)
- ZTE MF656A
- Vodafone Mobile Connect K3565
- Huawei K4505 (Vodafone Mobile Broadband)
- LTE Yota LiTE LU 156 4G - NOTE: May need nudged in some way out of storage mode. (e.g. boot delay, unplug/replug)
- Novatel EU850D (Mini PCIe)
- ZTE MF683 (May need CD-ROM disabled using AT+ZCDRUN=8 on another system first)
- ZTE MF622
- Ericsson H5321G / Lenovo FRU 60Y3297
- Ericsson F5521GW Gobi3000 / Lenovo
- Ericsson N5321 / Lenovo
  - May need “AT+CFUN=1” in the init string. Serial port varies from /dev/cuaU[0-3]
- ZTE MF915 LTE modem (T-Mobile)
- ZTE MF190 USB (1&1) using */dev/cuaU0.2*
- ZTE MF669 - May need “camcontrol eject da0” in shellcmd, uses */dev/cuaU0.2*

- ZTE MF830 - Can be switched from Ethernet to Modem by accessing the device's web interface, depending on preference.
- ZTE MF861
- ZTE MF985 - Branded as AT&T Velocity 2
- D-Link DWM-157 (3.75HSPA+)
- ONDA MT503HSA Type MF636 (requires eject mode switch, see below)
- Netgear LB1120 (US)
- Netgear LB1121 (US)
- Netgear LB2120 (US)
- Netgear LB1110 (EU)
- Netgear LB1111 (EU)
- And many others

If a modem **DOES WORK** but is not on the list - Please *submit a documentation update*.

If a modem **DOES NOT WORK** - post about it on the Netgate Forum for help, **do not contact Netgate asking for support or drivers**.

32.2.1 Modems reported to work as Ethernet devices

- Verizon (Pantech) 295 - Works, but fails if detached and reattached, must reboot.
- ZTE MF60 3g
- ZTE MF90
- ZTE MF823 - Defaults to 192.168.0.1, will need to be sure local system does not have an overlapping network.
- ZTE MF833V - Same
- ZTE MF915 LTE modem (T-Mobile)
- ZTE MF975S
- Huawei E3372h - Command to switch to the correct mode:

```
usb_modeswitch -v 12d1 -p 1f01 -c /usr/local/share/usb_modeswitch/12d1:1f01
```

  The command has to be executed every time it's detached and reattached, the interface has to be disabled and then enabled again.

- Huawei E5573 - Command to switch to the correct mode:

```
usb_modeswitch -v 12d1 -p 1505 -c /usr/local/share/usb_modeswitch/12d1:1505
```

  The command has to be executed every time it's detached and reattached, the interface has to be disabled and then enabled again.

- Huawei E8372h – See *Mode Switching* and #6226
- TP-LINK M7350

32.2.2 Modem variations reported to NOT work

These have the same model numbers as the above, but have different chips and may not be supported.

- Huawei E173s

```
#Before switching (USB mass storage)
DefaultVendor= 0x12d1
DefaultProduct=0x1c0b
#After switching into modem mode
TargetVendor=  0x12d1
TargetProductList="1c05,1c08"
```

- mPCIe: Sierra Wireless Gobi2000

32.2.3 Mode Switching

Some devices show up as a media device, such as *cd0*, in this case it may be possible to switch modes by executing a command:

```
camcontrol eject cd0
```

If that does switch the modem to the proper mode, it may be added as a *Executing Commands at Boot* using the full path:

```
/sbin/camcontrol eject cd0
```

`usb_modeswitch` is required in order to make certain devices switch to the correct mode.

In some cases, switching to the correct mode is too late, resulting in an error "Network interface mismatch" on the console. In this case, the additional command `&& sleep 10` can be added. For example:

```
/usr/local/sbin/usb_modeswitch -v 12d1 -p 1f01 -c /usr/local/share/usb_modeswitch/
↪12d1:1f01 && sleep 10
```

This package is available in the pfSense repository, but cannot be installed using the GUI package manager. It can be installed from a shell prompt using `pkg install usb_modeswitch`.

It's not recommended to use this method on a production firewall, as the method has not been tested officially.

32.3 Verizon UML290 Cellular Modem

These instructions are specifically for the Verizon 4G modem Pantech UML290.

First, find the phone number associated with the device. One way to find it is to install the VZAccess Manager on a Windows PC and then plug in the device and let it detect the phone number. I also connected from the Windows PC to ensure it was activated but I'm not sure if that is necessary.

Follow the instructions here for how to configure the interface in pfSense® software: *Configuring 3G modems*.

Some setups may need to modify the instructions because `#777` may not work. Instead try `*99***3#`. Also for **username** try `devicephone#@vzw4g.com` and for **password** put `vzw`.

32.4 Configuring a Verizon UML295 USB Modem

Connect the UML295 to a computer to do the initial setup and remove the initial SIM PIN.

Test the connection from the computer to ensure that there is 4G connectivity.

If the test succeeded, remove the modem from computer and plug it into the firewall running pfSense® software. There should be no PPP configuration necessary for this device in the firewall as it will register as a USB Ethernet device in the interface list.

Note: Reboot the firewall if the device does not show in the list for assignment.

Assign **UE0** as a DHCP interface, and it should show an IP address in the `192.168.32.0/24` subnet which then should be accessible from the LAN after configuring the necessary rules.

The gateway on the new interface can also be used by the firewall for failover. See *Multiple WAN Connections* for details.

CHAPTER

THIRTYTHREE

TROUBLESHOOTING

33.1 Troubleshooting Asymmetric Routing

Asymmetric routing happens when traffic between two nodes takes a different path in each direction (e.g. A->B->C, C->D->A), it can be a problem for TCP which has strict state tracking but often does not affect "stateless" protocols such as ICMP or UDP.

33.1.1 Common Scenario

What happens in most cases is this:

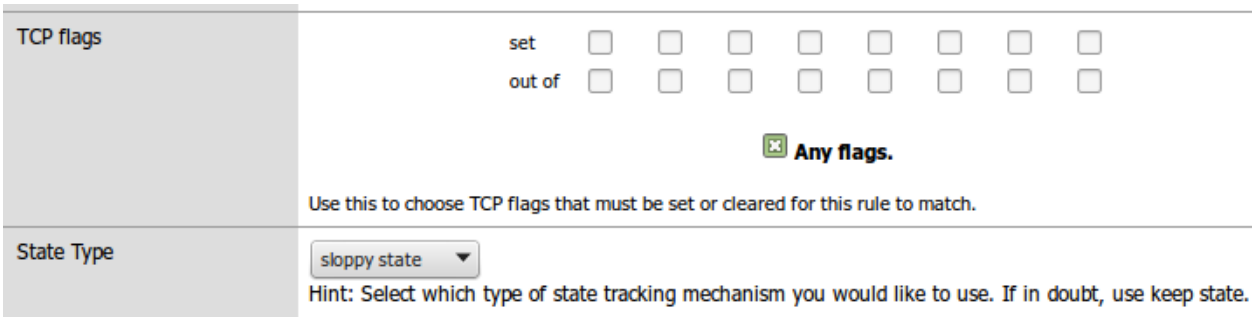
- Client sends a TCP SYN packet, which arrives to pfSense® software and gets a state table entry
- pfSense softwaresends back an ICMP redirect letting the client know to reach the target server via the alternate gateway
- Server sends back a TCP SYN+ACK packet by some other path that pfSense software doesn't see
- Client sends its ACK and further responses back by its other gateway that are not seen by pfSense software
- After 30 seconds, pfSense software removes its state table entry as the connection was never completed as observed by pfSense software
- Some time later, the client's ICMP redirect learned route expires and the client sends another packet back to pfSense software
- Since this packet is not starting a new connection, the packet is dropped, and the client gets disconnected since it now has no way to reach the destination.

33.1.2 Automatic Fix

The **Bypass firewall rules for traffic on the same interface** option located under **System > Advanced** on the **Firewall/NAT tab** activates rules for traffic to/from the static route networks which are much more permissive when it comes to creating states for TCP traffic and allowing it to pass. The rules allow any TCP packets, regardless of their flags, to create a state and also have the "*sloppy state*" type set which performs a less strict state match.

33.1.3 Manual Fix

The same rules may be created manually by adding one on the affected **interface tab** (e.g. LAN), and a second rule on the **Floating tab** using the same interface (LAN again) to match the traffic in the *out* direction. The rule must be set for a protocol of *TCP*, under **TCP flags** check *Any Flags*, and use a **State Type** of *Sloppy State*. The options for **TCP flags** and **State Type** can be found in the **Advanced Features** of the rules, under the normal options.



33.1.4 Alternate Causes

On occasion these issues can be caused by other factors that lead to asymmetric routing, such as issues with *route-to* or *reply-to*, both having to do with gateways on interface settings.

Defining gateways under **System > Routing** does not cause this, but rather these problems can come up when the gateway is improperly configured on the interface pages, **Interfaces > WAN**, **Interfaces > LAN**, and so on.

Gateway set when it should not be set

If a gateway is set on an internal interface, such as LAN, it can cause problematic behavior. Setting a gateway on an internal interface will tag that interface's outbound rules with *route-to*, and inbound rules with *reply-to* which will cause packets to be forwarded to the defined gateway rather than following their natural path. For WANs this is typically a good thing! For LANs it is not. Among other ill effects, it can lead to a loop of sorts where packets bounce between the firewall and the defined gateway, eventually being blocked or dropped when their TTL expires.

Gateway not set when it should be set

A gateway should usually be set on a WAN or other external-type interface settings (MPLS, IP VPN, etc.) In these cases the *reply-to* and *route-to* behavior is desired and likely required. If it is missing the packets may be blocked or dropped as they attempt to leave the wrong interface. A packet could enter via the alternate WAN, but the reply would leave by the default gateway. Similar to the effect seen when improperly using an *Interface Group* for WAN interfaces.

33.2 Troubleshooting Authentication

Testing authentication servers is possible using the tool located at **Diagnostics > Authentication**.

See also:

- *Authentication Servers*
- *External User Authentication Examples*
- pfSense Hangouts on Youtube to view the August 2015 Hangout on RADIUS and LDAP.

Testing user authentication is a simple process:

- Navigate to **Diagnostics > Authentication**
- Select an **Authentication Server**
- Enter a **Username**
- Enter a **Password**
- Click the **Test** button.

Note: This only performs a basic authentication test. Some special use cases, such as EAP, cannot be tested in this manner and may still fail when this test succeeds.

The firewall will attempt to authenticate the user against the chosen server, then it prints the results. The best practice is to try this at least once before attempting to use the server.

If the server returned a set of groups for the user, and the groups exist locally with the same name, the GUI prints the names of the groups in the test results.

If the firewall receives an error when testing authentication, double check the credentials and the server settings, then make any necessary adjustments and try again.

33.2.1 LDAP DN and Related Settings

For LDAP authentication servers, first ensure the base DN and similar settings match those configured on the LDAP server. Check the LDAP server for more information.

For **Base DN**, it's common to use the root of the LDAP tree but in most cases **Entire Subtree** must also be selected for the **Search Scope**.

Authentication Containers vary by LDAP implementations and setup. On Windows, it is commonly `CN=Users,DC=example,DC=com`, but it may vary. Try using an LDAP browser or similar software to locate the correct container.

LDAP path components are not case sensitive, so `CN=Administrator` is equivalent to `cn=administrator`.

33.2.2 Bind Credentials

When using authenticated (Not anonymous) binds, the username can be the short name (e.g. `DOMAIN\User` for AD) or a full LDAP specification for a user (e.g. `CN=administrator,CN=Users,DC=example,DC=com`).

Tip: The full DN of a Windows AD bind user can be found by navigating to the user in **ADSI Edit** found under **Administrative Tools** on the Windows server.

For a production setup, an unprivileged user should be used for binding if possible, and not AD Administrator-level account. Such an unprivileged user may need sufficient permissions to attempt binding as other users and access the LDAP directory.

Another common mistake with group membership is not specifying *Entire Subtree* for the **Search Scope Level**.

33.2.3 Active Directory Group Membership

Depending on how the Active Directory groups were made, the way they are specified may be different for things like Authentication Containers and/or Extended Query. For example, a traditional user group in AD is exposed differently to LDAP than a separate Organizational Unit.

ADSI Edit found under **Administrative Tools** on the Windows server can locate the DN for a given group.

33.2.4 Extended Query

The most common mistake with **Extended Query** is that the given directive fails to include both the item to be searched as well as how, such as:

```
memberOf=CN=VPNUsers,CN=Users,DC=example,DC=com
```

Note that in the above example the DN of the group is given along with the restriction (`memberOf=`).

33.2.5 Connection-Related Issues (non-SSL/TLS)

Make sure that the LDAP server is listening on the expected port, and that connectivity to the LDAP server network is functional.

Performing a packet capture filtered on the LDAP server IP address and port will help track down the problem. Tools such as **Wireshark** can interpret LDAP packets and help diagnose queries and failures. See *Troubleshooting via Packet Captures*.

33.2.6 Connection-Related Issues (SSL/TLS)

By far the most troublesome connection type is LDAP+SSL/TLS (ldaps) due to its strict security and validation standards.

Hostname Required

When connecting to LDAP with SSL/TLS, the hostname given for the server is also used to verify the server certificate. The server certificate SAN entries and/or CN **must include its hostname**, and that **hostname must resolve to the LDAP server IP address**, e.g. `CN=ldap.example.com`, and `ldap.example.com` is `192.168.1.5`.

If an IP address has been entered for the hostname of the LDAP server, it will not work unless that IP address happens to also be the CN or a SAN of the server certificate.

If this must be worked around, it is possible to create a DNS host override in the DNS forwarder for `<common name of the cert>.<firewall domain name>`. That assumes that the CN is in a format that could actually be a hostname, or that the hostname in question is present in a SAN entry on the server certificate.

Use the Correct Port

When using port `636` for SSL/TLS, the firewall uses an `ldaps://` URL, not STARTTLS. Ensure that the LDAP server is listening on the correct port with the correct mode. For STARTTLS, use port `389`.

Ensure CA Matches

The most important factor in making sure that it is possible to communicate with the LDAP server over SSL/TLS is that the correct CA certificate has been imported into the firewall, and is chosen on the LDAP settings. The key is not required, only the CA certificate.

Nested CAs

If the LDAP server certificate CA is part of a chain, or there is an intermediate CA, every CA certificate in the chain must imported into the Certificate Manager.

Other Cert/CA Issues

Confirm that the certificates are otherwise valid, for example they are not expired or set to be valid in the future.

33.2.7 Troubleshooting via Server Logs

Authentication failures are typically logged by the target server (FreeRADIUS, Windows Event Viewer, etc), assuming the request is making it all the way to the authentication host. Check the server logs for a detailed explanation why a request failed. The system log at **Status > System Logs** may also contain information that hints at a resolution.

33.2.8 Troubleshooting via Packet Captures

Packet captures (*Packet Capturing*) can be invaluable for diagnosing errors as well. If an unencrypted method (RADIUS, LDAP without SSL/TLS) is in use, the actual password being used may not be visible but enough of the protocol exchange can be seen to determine why a request is failing to complete. This is especially true when a capture is loaded in Wireshark, which can interpret the responses, as seen in Figure *Sample LDAP Failure Capture*.

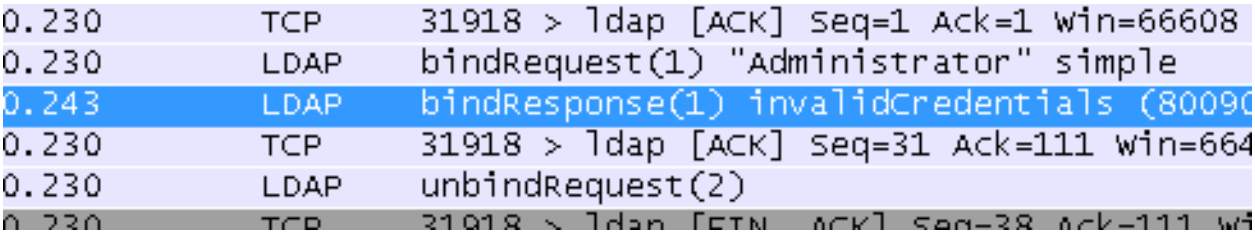


Fig. 1: Sample LDAP Failure Capture

33.3 Troubleshooting Boot Issues

The errors in this article would happen during bootup, typically the first boot either from the install media or immediately after installation. If the system was up and running but then developed a boot issue with no changes in the software/OS, it isn't likely to be related.

33.3.1 Booting with an alternate console

To boot a different console, first get to a loader prompt. Either choose the menu option from the boot menu, or when **Hit [Enter] to boot immediately, or any other key for command prompt.** is seen during the boot process, press space or another key.

Once at the loader prompt, type the following to boot with the serial console active:

```
set console=comconsole
boot -v
```

Similarly, if a serial memstick image is used that prefers the serial console, the video console can be used instead as follows:

```
set console=vidconsole
boot
```

33.3.2 Booting from USB

- If the boot stops with a mountroot error while booting off the installation disc, usually with USB CD/DVD drives, escape to the loader prompt and run the following:

  ```
  set kern.cam.boot_delay="10000"
  boot
  ```

- On 2.0 this is on the boot menu - option #3 to boot from USB devices.

  At which point the boot will continue normally and a normal installation will be possible.

  If running permanently from a medium that requires this delay, edit /boot/loader.conf.local and insert the following line:

  ```
  kern.cam.boot_delay="10000"
  ```

- If booting fails from a USB 3.0 port and the above does not help, try a USB 2.0 port with the same delay settings.

33.3.3 GPT Boot Issues

Some systems may fail to boot a 2.4 memstick because they do not fully support booting from GPT or they are particular about the layout or other attributes.

In these cases, the memstick can be modified using another firewall running pfSense® software version 2.3 or later, or with a FreeBSD 10.x or later system.

Insert the memstick into the pfSense or FreeBSD device and check the system log or dmesg output to find the device name, such as **da1**.

First, try to set the partition on the memstick active:

```
gpart recover da1
gpart set -a active da1
```

Remove the memstick and attempt to boot the memstick on the target hardware. If it works, be sure to only use compatible options in the installer. In this case, if the BIOS requires an active GPT partition then either of the following installer choices will work:

- **Auto (UFS)** requires the Partition Scheme **BSD (BSD Labels)**
- **Auto (ZFS)** requires the Partition Scheme **GPT + Active (BIOS)**

Other platforms may need more changes to the memstick, including:

```
gpart bootcode -b /boot/pmbr -p /boot/gptboot -i 2 da1
gpart set -a bootme -i 2 da1
```

In this case the hardware may not be capable of booting ZFS, but use the same option above for UFS with BSD Labels which should work.

In either case, depending on the BIOS and hardware capabilities, similar modifications may be made to the installation target disk to use other partition schemes not listed above, but that has not been tested internally.

33.3.4 BIOS/Disk Errors

- If after installation, a "cannot load kernel" error is observed, or the firewall hangs at the spinner (/):
  - Make sure BIOS is up to date
  - Reinstall but do not check the packet mode option during boot block installation process
  - Set the HDD mode in BIOS to LBA (don't use "auto", the detected geometry is different if it is set to auto and it fails)
  - Set the HDD mode in BIOS to CHS if the above fails
  - Set AHCI mode in the BIOS
- Try using multiple partitions, one small one (~4GB) for / and another for /usr using the rest of the disk.
- If the installer fails to start correctly (system reboots), try the following (attempt each substep, then rerun the installer each time):
  - Make sure BIOS is up to date
  - Change the hard drive ribbon/SATA cable
  - Force a slower speed in the bios for the channel
  - Boot from another disk and run: (note that ad0 is the first ata hard drive):

    ```
    # dd if=/dev/zero of=/dev/ad0 bs=8k count=16
    # fdisk -I ad0
    ```

- If all those fail...
  - Partition from a FreeBSD Live CD or Linux
  - Different hard drive

33.3.5 CD Errors

- If the installer disc starts to boot but hundreds of errors are printed, try:
  - Make sure BIOS is up to date
  - Use a USB memstick instead
  - Burn the CD at a slower speed
  - Change the CD-ROM Reader ribbon cable
  - Different CD-ROM Reader

33.3.6 Boot Blocks/Loader Issues

- If a read error occurs during boot, please see this Boot Error.
- If FreeBSD will boot but not pfSense, try booting from a FreeBSD Live CD and running the following (More Info):

```
# fdisk -B -b /boot/boot0 /dev/ad0
# bsdlabel -B /dev/ad0s1
```

- (note that ad0 is the first ata hard drive)

33.3.7 Vendor-Specific Issues

- Certain Dell Blade servers may hang at boot if the system's virtual USB media is enabled. Disable the virtual media in the BIOS and then it should boot normally.
- Certain systems running Hyper-V on AMD processors may need to do the following:
  - Escape to the loader prompt during bootup and run:

```
set hw.clflush_disable=1
boot
```

    At that point, boot the rest of the way and install pfSense. After installation, add the following line to /boot/loader.conf.local:

```
hw.clflush_disable=1
```

33.3.8 Alternate Boot Managers

GAG or Smart BootManager may be used.

If all else fails, Netgate offers Netgate TAC and hardware through the Netgate Store that has been pre-loaded with pfSense software.

33.3.9 Disabling ACPI

Some hardware and/or BIOS implementations have incompatibilities with FreeBSD and ACPI. In these cases, ACPI may need to be disabled in order to boot or install successfully.

The boot menu displayed when the system starts has a choice to disable ACPI. Choose that option when booting to disable it temporarily and then follow the loader.conf.local suggestion below.

Disable in Loader

On some installations, such as those with a serial console, the boot menu is not displayed and the change must be made at the loader prompt.

At the Hit [Enter] to boot immediately, or any other key for command prompt.

At this point hit any key and a prompt will be displayed. Then type:

```
set hint.acpi.0.disabled=1
boot
```

Make it Persistent

After the installation, add the following line to /boot/loader.conf.local by running the following command from a shell prompt or exec.php:

```
echo "hint.acpi.0.disabled=\"1\"" >> /boot/loader.conf.local
```

Or use the Diagnostics>Edit File function to open /boot/loader.conf.local and add the following line:

```
hint.acpi.0.disabled="1"
```

Then save. It will be applied at the next boot.

33.3.10 Disable DMA for IDE drives

The hardware in use may not be capable of using DMA transfers. In such cases, DMA errors will be observed when installing pfSense. Disabling DMA support in BIOS might work. Another option is to disable DMA support at boot time. This will slow a DMA capable system down. It should only be used when DMA errors are encountered when accessing the hardware.

To disable DMA:

After powering on the system, the following message appears: Hit [Enter] to boot immediately, or any other key for command prompt.

At this point hit any key and a prompt will be presented.

pfSense 2.1 and earlier

To disable DMA for hard drives and compact flash:

```
set hw.ata.ata_dma=0
```

To disable DMA for optical drives:

```
set hw.ata.atapi_dma=0
```

After the installation, add the following line to */boot/loader.conf.local*:

To disable DMA for hard drive(s):

```
hw.ata.ata_dma=0
```

To disable DMA for optical drives:

```
hw.ata.atapi_dma=0
```

It is possible that DMA may need to be disabled for both hard drive(s) and optical drives.

pfSense 2.2 and later

The method by which DMA is disabled has been changed on pfSense 2.2 and later due to changes in the underlying disk driver structure on FreeBSD.

To disable DMA from a loader prompt, use:

```
set hint.ata.0.mode=PIO4
```

To make the change permanent, add the following line to */boot/loader.conf.local*:

```
hint.ata.0.mode=PIO4
```

If there are multiple ATA controllers, the controller ID may need to be set higher (e.g *1* or *2*) or it may need set for multiple controllers. In that case, use additional lines with the same setting that vary only by controller ID.

Possible modes for that setting include:

> BIOSDMA, PIO0 (alias BIOSPIO), PIO1, PIO2, PIO3, PIO4, WDMA2, UDMA2 (alias UDMA33), UDMA4, (alias UDMA66), UDMA5 (alias UDMA100) and UDMA6 (alias UDMA133).

33.3.11 Disable Write Caching

In some cases a disk or controller may need write caching disabled to ensure that data is written to disk immediately and not held in a cache.

To disable write caching at boot time (if the system will not boot or install otherwise), use a loader prompt:

After powering on the system, the following message appears: Hit [Enter] to boot immediately, or any other key for command prompt.

At this point hit any key and a prompt will be presented.

pfSense 2.1 and earlier

To disable write caching for all ATA drives:

```
set hw.ata.wc=0
```

After the installation, add the following line to */boot/loader.conf.local*:

To disable write caching for all ATA drives:

```
hw.ata.ata_dma=0
```

pfSense 2.2 and later

The method by which write caching is disabled has been changed on pfSense 2.2 and later due to changes in the underlying disk driver structure on FreeBSD.

To disable write caching from a loader prompt, use:

```
set kern.cam.ada.write_cache=0
```

To make the change permanent, add the following line to */boot/loader.conf.local*:

```
kern.cam.ada.write_cache=0
```

33.3.12 "Fake" RAID cards with a GRAID error

Certain "fake" RAID cards, driver/software-based RAID adapters that are not true hardware RAID, may fail to mount properly with the following error:

```
Root mount waiting for: GRAID
mountroot>
```

Another symptom can be that "Intel RAID" messages are shown during the boot sequence, and typing ? at the mountroot prompt it only shows the drive itself and no partitions:

```
Mounting from ufs:/dev/ada0s1a failed with error 19
mountroot> ?
[...]
ada0
```

- Escape to a loader prompt during bootup and run:

  ```
  set kern.geom.raid.enable="0"
  boot
  ```

- After a successful install/boot, add that settings permanently to */boot/loader.conf.local*:

  ```
  kern.geom.raid.enable="0"
  ```

33.3.13 Conflicting Hardware

- If the system hangs right after detecting the hard drives, and the floppy drive light is on, turn off floppy support in the BIOS.
- If the system stops with an error such as:

```
run_interrupt_driven_hooks: still waiting after 60 seconds for xpt_action
```

  Disable any Firewire/1394 controllers and USB Card Readers in the BIOS.

33.4 Troubleshooting "No buffer space available" Errors

On occasion traffic on a NIC may have trouble getting out with an error similar to:

```
ping: sendto: No buffer space available
```

Or:

```
dpinger WANGW x.x.x.x: sendto error: 55
```

The most common causes of this are:

- No route to the target network (or no default route)
- Missing link route for a local target
- Stale state in pf sending the connection out an invalid path (reset states)
- Network memory buffer exhaustion - See *Tuning and Troubleshooting Network Cards*
- Faulty NIC and/or driver issue
  - Sometimes resetting the NIC can bring it back again:

```
# ifconfig em3 down; ifconfig em3 up
```

- Faulty cable
- Traffic shaping (ALTQ or Limiters) dropping the packet
- Virtual NIC being throttled by the hypervisor or host, such as an AWS instance using more throughput than an instance size can support
  - In this case, change the throttling in the host (not guest) or upgrade to a larger instance/higher tier on a hosted platform such as AWS.
- Virtual NIC being disconnected/disabled in certain hypervisors
- An otherwise overloaded NIC exhausting its send/recv buffers
- Other various switch/buffer/connectivity issues

Trying to bounce the NIC with *ifconfig* is the easiest thing to try first. After that, save/apply the interface settings on each interface (or at least WANs and the LAN in question). Check/(re)set the default route if it has been lost. *Reset States*. Replacing the cable may also help. Removing traffic shaping if it is enabled is also a good test.

Otherwise investigate the traffic on the NIC and look for other buffer-related causes. Seek help from Netgate TAC for assistance in diagnosing the issue, or post on the forum/mailing list.

33.5 Troubleshooting Captive Portal

This section contains troubleshooting tips for the most common problem with captive portal.

33.5.1 Authentication failures

Authentication failures are normally the result of users entering an incorrect username or password. In the case of RADIUS authentication, these can occur because of connectivity problems to the configured RADIUS server(s), or problems on the RADIUS server itself. Check the RADIUS server logs for indications of why access was denied, and ensure the firewall can communicate with the RADIUS server.

For local users, if the option to require the Captive Portal login privilege is enabled, ensure the users have the privilege directly or are members of a group with the privilege.

33.5.2 Captive Portal Does not Redirect

If clients are not being redirected to the portal page when attempting to browse on an interface with captive portal enabled, it's most always one of the following causes:

DNS resolution not functioning Clients on the captive portal interface must either be using the DNS resolver or forwarder on pfSense® software, on the IP address of the interface where the client resides (which is the default configuration), or if using another IP address for DNS, it must be in an allowed IP address entry. If DNS fails, the browser never issues the HTTP request, hence it cannot be intercepted and redirected.

Firewall rules on the captive portal interface do not allow the initial HTTP request If the user is trying to browse to google.com, but HTTP connections are not allowed to google.com, the HTTP request will be blocked and hence cannot be redirected. Under **Firewall > Rules**, on the interface where captive portal is enabled, the traffic to be redirected must be allowed to pass. This is most commonly HTTP to any destination.

The client has an HTTPS home page The request must be to an HTTP site in order for the portal to redirect the client. If HTTPS is enabled for the portal, this may still work but it depends upon the client browser or operating system automatic portal detection to work.

33.5.3 Apple devices are unable to load the portal page or login

Certain versions of Safari on iOS do not properly handle the login form for the Captive Portal page. The most common resolution is to disable autofill for forms in Safari on iOS.

In some cases, Apple devices will not automatically prompt for a Captive Portal login or test for its presence if the wireless network uses encryption. In these cases, manually open a browser and navigate to an HTTP site to get the login redirect.

There have also been reports that on older version of OS X, a Mac would refuse to load any HTTPS sites, including an HTTPS portal, until it could load a CRL and OSCP URL for the certificate. This has been fixed in current versions of OS X.

Some users have had to add `www.apple.com` to their allowed hostnames so that Apple's call to their test page succeeds.

33.5.4 Port Forwards Behind Portal Only Work When Target Logs In

This is a side effect of how the portal operates. No traffic is allowed to reach a host behind the portal unless it has been authenticated or passed through the portal. If a port forward must always work to a device behind the portal, then it must be setup to bypass the portal with either a Pass-through MAC entry (*MAC Address Control*) or an Allowed IP Address entry (*Allowed IP Address*) to allow traffic *To* the target.

33.5.5 Captive Portal Rules

Captive Portal uses `ipfw` under the hood. `ipfw` is a program performing packet filtering. When having issues with the captive portal, it is possible to list `ipfw` rules for debugging.

To list all ipfw rules, which includes rules for Captive Portal in general as well as zone specific tables, run:

```
# ipfw show
```

IPFW Tables

Show all tables:

```
# ipfw table all list
```

The **<name>_auth_up** table holds authenticated/allowed clients for a zone. This table allow traffic from clients to enter the interface. For example, a zone called "myzone" would contain this table:

```
# ipfw table myzone_auth_up list
```

The **<name>_auth_down** table holds authenticated/allowed clients for a zone. This table allow traffic to clients to exit the interface. For example, a zone called "myzone" would contain this table:

```
# ipfw table myzone_auth_down list
```

See `/etc/inc/captiveportal.inc` for information on other tables, these include tables for host/MAC bypass entries and other necessary controls.

See also:

For assistance in solving problems, post on the Captive Portal category of the Netgate Forum.

33.6 Troubleshooting Cisco VPN Pass Through

If trouble is encountered when attempting a connection from an internal Cisco VPN client to an external host, (e.g. a workstation with the Cisco client is trying to get out through a pfSense® firewall to connect to a "foreign" site), then try the following.

33.6.1 Workaround

- In the Cisco VPN client software, Modify the connection and turn off transparent tunneling completely in the **Transport** tab.
- In the pfSense webGUI, under **Firewall > NAT** on the **Outbound** tab:
  - Enable **Manual Outbound NAT**.
  - Remove any NAT rules that perform static port NAT on udp/500.

33.7 Troubleshooting Network Connectivity

The following list covers nearly every cause of outbound connectivity failure in common usage scenarios. Each test assumes the items above it have been checked.

33.7.1 WAN Interface

- Check that the WAN IP address is correct (**Interfaces > WAN**)
  - Using the wrong address could cause a failure of the ISP to deliver traffic to/from the firewall, among other issues
- Check that the WAN IP address has the correct subnet mask (**Interfaces > WAN**)
  - An improper subnet mask such as /1 could cause connectivity issues to large portions of the Internet, using /32 for a mask can prevent the gateway from being found/used
- Check that WAN has a gateway and that the gateway IP is correct (**Interfaces > WAN**)
  - This will interfere with automatic outbound NAT and route-to/reply-to handling
- Check that the WAN gateway is set as default (**System > Routing**)
  - Without a default gateway traffic has no exit path
- Check that the WAN gateway shows Online (**Status > Gateways**)
  - If it is not, verify the WAN settings and gateway settings, or use an alternate monitor IP
- Verify that the defined WAN gateway is actually the default (**Diagnostics > Routes**)
  - Some other source such as a VPN may have changed the default gateway

33.7.2 LAN Interface

- Check that the LAN IP address is correct (**Interfaces > LAN**)
  - Using an invalid IP address (e.g. .0 or .255 in a /24) will cause problems reaching addresses locally and will not work properly.
- Check that the LAN subnet mask is correct (**Interfaces > LAN**)
  - Using an incorrect subnet mask, such as /32, will prevent other hosts in LAN from finding the LAN to use as a gateway and vice versa
- Check that LAN does NOT have a gateway set (**Interfaces > LAN**)
  - This will interfere with automatic outbound NAT

- Check that LAN does NOT have "Block Private Networks" set (**Interfaces > LAN**)
  - Should be obvious
- Check that LAN does NOT have "Block Bogon Networks" set (**Interfaces > LAN**)
  - See above

33.7.3 Firewall/Rules

- Check the firewall log for blocked connections from the LAN (**Status > System Logs**, **Firewall** tab)
  - If blocks are observed, check the rule that blocked and adjust rules accordingly (**Firewall > Rules**, **LAN** tab)
- Check that the LAN rule allows all protocols, or at least TCP and UDP ports for reaching DNS and HTTP/HTTPS, and allows ICMP for testing. (**Firewall > Rules**, **LAN** tab)
  - Not allowing UDP would make DNS fail, among other things. Similarly, on a DNS rule, using UDP only and not TCP/UDP will cause larger queries to fail.
  - Not allowing ICMP would cause ping to fail, but other protocols may work
  - Not allowing TCP would cause HTTP, HTTPS, and other protocols to fail.
- Check that the LAN rule allows to a destination of *any* (**Firewall > Rules**, **LAN** tab)
  - Traffic going to the Internet will need an "*any*" destination. Using the wrong destination would not allow traffic to reach the Internet (e.g. "*WAN net*" which is only the subnet of the WAN interface, **NOT** the Internet.)
- Check that the LAN rule does not have an improper gateway set (**Firewall > Rules**, **LAN** tab)
  - If it is set to leave by some other (possibly broken) non-WAN gateway it would cause the connections to fail

33.7.4 Outbound NAT

- Check **Outbound NAT**, ensure it is set for *Automatic Outbound NAT* unless Manual is required (**Firewall > NAT**, **Outbound** tab)
  - Incorrect NAT settings will prevent traffic from reaching WAN
- Check Manual Outbound NAT rules, if in use, to ensure that the source of local traffic is matched
  - Incorrect NAT settings will prevent traffic from reaching WAN

33.7.5 Diagnostic Tests

- Check connectivity from the firewall itself: Try to ping *8.8.8.8* (**Diagnostics > Ping**)
  - If this does not work, ensure proper WAN settings, gateway, etc.
- Check DNS: Try to lookup *pfsense.org* (**Diagnostics > DNS Lookup**)
  - If this does not work, fix/change the DNS servers on **System > General**
- Test NAT: Try to ping *8.8.8.8* (**Diagnostics > Ping**) using *LAN* as the **Source Address**
  - If this fails but the other tests work, then the problem is likely Outbound NAT (See the WAN/LAN gateway checks above)

33.7.6 Client Tests

- Test if the client can ping the LAN IP of the firewall
  - If this fails, check the LAN rules, client IP/subnet mask, LAN IP/subnet mask, etc.
- Test if the client can ping the WAN IP of the firewall
  - If this fails, check the client's subnet mask and gateway
- Test if the client can ping the WAN Gateway IP of the firewall
  - If this fails, check the client's subnet mask and gateway, and double check Outbound NAT on the firewall
- Test if the client can ping an Internet host by IP address (e.g. *8.8.8.8*)
  - If this fails, check the client's subnet mask and gateway, and triple check Outbound NAT on the firewall
- Test if the client can ping an Internet host by Host name (e.g. *www.google.com*)
  - If this fails, check the client's DNS settings, and/or the DNS Forwarder on the firewall (**Services > DNS Forwarder**, **Diagnostics > DNS Lookup**)

33.7.7 Miscellaneous Additional Areas

- If Captive Portal is enabled, temporarily disable it (**Services > Captive Portal**).
  - See *Captive Portal Troubleshooting* if that helps.
- Check for packages such as Squid that might interfere, disable them if necessary
  - Improperly configured proxies would allow certain traffic such as ICMP ping to work but might prevent access to HTTP and/or HTTPS sites.

33.8 Troubleshooting GUI Connectivity

If the WebGUI is not accessible from the LAN, the first thing to check is cabling. If the cable is a hand-made cable or *shorter* than 3 feet/1 meter, try a different cable. If the client PC is directly connected to a network interface on the firewall, a crossover cable may be needed on older hardware that does not have Auto-MDIX support on its network cards. This is not a concern on gigabit or faster hardware.

Once there is a link light on both the client network card and the firewall LAN interface, check the TCP/IP configuration on the client PC. If the DHCP server is enabled on the firewall, as it is by default, ensure that the client is also set for DHCP. If DHCP is disabled on the firewall, hard code an IP address on the client residing in the same subnet as the firewall LAN IP address, with the same subnet mask, and use the firewall LAN IP address as the gateway and DNS server.

If the cabling and network settings are correct, the client will be able to ping the LAN IP address of the firewall. If the client PC can ping, but not access the WebGUI, there are still a few more things to try. First, if the error on the client PC is a connection reset or failure, then either the server daemon that runs the WebGUI is not running or the client is attempting to use the wrong port. If the error is instead a connection timeout, that points more toward a firewall rule.

If the client receives a connection timeout, refer to *Troubleshooting Access when Locked Out of the Firewall*. With a properly configured network connection, this shouldn't happen, and that section offers ways to work around firewall rule issues.

Double check that WAN and LAN are not on the same subnet. If WAN is set for DHCP and is plugged in behind another NAT router, it may also be using `192.168.1.1`. If the same subnet is present on WAN and LAN, unpredictable results may happen, including not being able to route traffic or access the WebGUI. When in doubt, unplug the WAN cable, reboot the firewall, and try again.

If the client receives a connection reset, first try to restart the WebGUI server process from the system console, typically option 11, followed by option 16 to restart PHP-FPM. If that does not help, start a shell from the console (option 8), and type:

```
# sockstat | grep nginx
```

The firewall will return a list of all running `nginx` processes, and the port upon which they are listening, like this:

```
root     nginx      41948 5  tcp4   *:443                 *:*
root     nginx      41948 6  tcp6   *:443                 *:*
root     nginx      41948 7  tcp4   *:80                  *:*
root     nginx      41948 8  tcp6   *:80                  *:*
```

In that output, it shows that the process is listening on port 443 of each interface on both IPv4 and IPv6, as well as port 80 for the redirect, but that may vary based on the firewall configuration.

Try connecting to the firewall LAN IP address by using that port directly, and with both HTTP and HTTPS. For example, if the LAN IP address was `192.168.1.1`, and it was listening on port `82`, try `http://192.168.1.1:82` and `https://192.168.1.1:82`.

33.9 Troubleshooting Offline DHCP Leases

The **Status > DHCP Leases** page only reports systems as "online" if the MAC address for a given system appears in the pfSense® firewall's ARP table. This can be verified by checking **Diagnostics > ARP Table**. Systems that have not communicated with or via the firewall in the past few minutes will appear as offline.

To check a system, try to ping it from **Diagnostics > Ping**. Even if the system does not respond to ping, that action will cause the system to appear in the ARP table if it is on the network, and would thus show online in the DHCP Leases list.

The *Arping Package* may also be of interest. It is available under **System > Packages**. The *Nmap package* can also be used to perform an ARP-based subnet scan to locate online hosts even if they don't respond to ICMP.

33.10 Troubleshooting DHCPv6 Client XID Mismatches

An IPv6 WAN configured to obtain is address via DHCPv6 can suddenly find itself without an IPv6 address if the transaction ID for the IPv6 DHCP client does not match.

When this happens, a log message similar to the following may appear in the DHCP and/or System logs:

```
dhcp6c[xxxxx]: client6_recvadvert: XID mismatch
```

When this happens, the most common cause is having multiple running copies of the DHCPv6 client (`dhcp6c`) on the same interface. Both clients send out a request with different transaction IDs and then get confused by the responses.

When this happens, the quickest way to ensure the clients are reset is to kill the duplicates and start the client again.

From the shell or from **Diagnostics > Command Prompt**, first check for duplicate clients:

```
# ps uxawww | grep dhcp6c
root xxxxx 0.0 0.0 5780 1488 ?? INs Sat09PM 0:00.90 /usr/local/sbin/dhcp6c -d -c /var/
↪etc/dhcp6c_wan.conf -p /var/run/dhcp6c_re1.pid re1
root xxxxy 0.0 0.0 5780 1524 ?? Is Tue07AM 0:00.30 /usr/local/sbin/dhcp6c -d -c /var/
↪etc/dhcp6c_wan.conf -p /var/run/dhcp6c_re1.pid re1
```

Once it has been confirmed that a duplicate client is running, `kill` them:

```
# killall -9 dhcp6c
```

Then navigate to **Interfaces > WAN**, click **Save**, then click **Apply Changes**.

The WAN should now have its IPv6 address once again.

33.11 Troubleshooting DMA and LBA Errors

33.11.1 Non-Fatal Errors

CF NID Not Found Error

Some newer Compact Flash cards are giving users trouble, in particular the Sandisk 4GB 30MB/s cards seem to be problematic.

The cards may function properly, but will often generate an error in the logs such as:

```
ad0: FAILURE - READ status=51<READY,DSC,ERROR> error=10<NID_NOT_FOUND> LBA=7813119
ad0: FAILURE - READ status=51<READY,DSC,ERROR> error=10<NID_NOT_FOUND> LBA=7813119
```

This normal and safe to ignore. It has nothing to do with image size, a 1 GB image can be used on a 4 GB card and they still do that. From what I've been able to gather on that, the cards report they have a larger usable size than they do, and when the FreeBSD ATA code initializes it tries to read the end of the disk even if its partition is nowhere near that, which the CF is reporting doesn't actually exist, which is what that log message says.

> according to ATA-7 specification, Section 6.59.6 is: "IDNF shall be set to one if a user-accessible address could not be found. IDNF shall be set to one if an address outside of the range of user-accessible addresses is requested if command aborted is not returned." FreeBSD labels this bit as NID_NOT_FOUND

Summarized from this forum post.

SETFEATURES Error

Newer snapshots and releases of pfSense® software (after 2.0.1) also attempt to setup APM for an ATA hard drive at boot. Sometimes the detection yields a non-fatal error when trying to find if the value is supported, or when the drive claims it's supported but can't set it. That error is:

```
ad0: FAILURE - SETFEATURES 0x85 status=41<READY,ERROR> error=4<ABORTED>
```

It should only be seen once at bootup on such systems, and can safely be ignored.

33.11.2 Fatal Errors

The following errors would indicate more serious problems such as a faulty HDD/CF, faulty cable/controller, a faulty CF/SATA/IDE converter, a device out of space, or possibly that *DMA needs disabled on that combination of hardware*.

```
ad0: FAILURE - READ_DMA status=51<READY,DSC,ERROR> error=40<UNCORRECTABLE> LBA=3919
```

```
ad0: WARNING - READ_DMA UDMA ICRC error (retrying request) LBA=207
ad0: WARNING - READ_DMA UDMA ICRC error (retrying request) LBA=207
ad0: FAILURE - READ_DMA status=51<READY,DSC,ERROR> error=84<ICRC,ABORTED> LBA=207
g_vfs_done():ad0s1a[READ(offset=65536, length=8192)]error = 5
```

```
ad0: FAILURE - WRITE_DMA status=51<READY,DSC,ERROR> error=4<ABORTED> dma=0x06␣
↪LBA=1129359
```

33.11.3 Other Errors

The following errors have been observed on certain hardware platforms running pfSense 2.2. While they do not appear to be fatal, the cause appears to be a disk driver issue in FreeBSD (9.2 and later) and it may degrade performance. If these errors are encountered, attempt to reproduce the problem with a stock FreeBSD 10.1 or later installation and report the problem directly to FreeBSD.

```
(ada0:ata0:0:1:0): READ_DMA. ACB: c8 00 3f 95 07 40 00 00 00 00 08 00
(ada0:ata0:0:1:0): CAM status: ATA Status Error
(ada0:ata0:0:1:0): ATA status: ff (BSY DRDY DF SERV DRQ CORR IDX ERR), error: 00 ()
(ada0:ata0:0:1:0): RES: ff 00 46 95 07 00 00 00 00 00 00
(ada0:ata0:0:1:0): Retrying command
```

```
(ada0:ata0:0:1:0): WRITE_DMA. ACB: ca 00 ef e8 1a 40 00 00 00 00 08 00
(ada0:ata0:0:1:0): CAM status: ATA Status Error
(ada0:ata0:0:1:0): ATA status: ff (BSY DRDY DF SERV DRQ CORR IDX ERR), error: 00 ()
(ada0:ata0:0:1:0): RES: ff 00 ef e8 1a 00 00 00 00 08 00
(ada0:ata0:0:1:0): Retrying command
```

This appears to further be limited to onboard CF-to-SATA sockets. Using another disk type (mSATA, SATA, etc) may also be a viable workaround.

33.12 Troubleshooting DNS Resolution Issues

Inside the WebGUI, navigate to **Diagnostics > Ping** and enter in the ISP gateway address. The gateway address is listed on **Status > Interfaces** for the WAN interface and under **Status > Gateways**.

If the gateway is unknown, try another known-valid address such as `8.8.8.8`. If the firewall is able to ping that address and receive a response, then repeat that same ping test from the client PC. Open a command prompt or terminal window, and ping that same IP address.

If the client can ping by IP address, then try to ping a web site by name such as `www.google.com`. Try it from the firewall GUI and from the client PC. If the IP ping test works, but the name test fails, then there is a problem with DNS resolution. See Figure *Testing Connectivity for Bogon Updates* for an example.

If DNS resolution does not work on the firewall, first check which DNS service is enabled on the firewall and how it is configured. By default, pfSense® software is configured to use the DNS Resolver in a mode that does not require any specific DNS servers. It queries the root servers and other authoritative servers directly. Older installations and

upgraded installations default to the DNS Forwarder, which requires DNS Servers to be entered under **System > General Setup** or to be acquired from a dynamic WAN such as DHCP or PPPoE. The DNS Resolver can also operate in this mode if **Enable Forwarding Mode** is activated in its settings.

If the DNS Resolver is active but the firewall is unable to resolve hostnames, the problem is usually a lack of working WAN connectivity. Aside from that, one possibility is that the WAN or upstream network gear does not properly pass DNS traffic in a way that is compatible with DNSSEC. Disable DNSSEC in the Resolver options to see if that allows resolution to function. It is also possible that the ISP filters DNS requests and requires the use of specific DNS servers. In that case, configure DNS servers and then activate forwarding mode or switch to the DNS Forwarder.

The firewall DNS server settings are under **System > General Setup**, and are also visible at **Status > Interfaces**. Check with ping to be sure these DNS servers are reachable. If the firewall can reach the gateway address at the ISP, but not the DNS servers, contact the ISP and double check those values. If the DNS servers are obtained via DHCP or PPPoE and the firewall cannot reach them, contact the ISP. If all else fails, consider using Google's public DNS (8.8.8.8, 8.8.4.4) name servers on the firewall instead of those provided by the ISP.

If DNS works from the firewall but not from a client PC, it could be the DNS Resolver or Forwarder configuration on the firewall, the client configuration, or firewall rules. Out of the box, the DNS Resolver handles DNS queries for clients behind the firewall. If the client PCs are configured with DHCP, they will receive the IP address of the firewall interface to which they are connected as a DNS server, unless that is manually changed. For example, if a PC is on the LAN interface, and the firewall LAN IP address is `192.168.1.1`, then the client DNS server should also be `192.168.1.1`. If the DNS Resolver and DNS Forwarder are disabled, adjust the DNS servers which get assigned to DHCP clients under **Services > DHCP Server**. Normally when the DNS Resolver and DNS Forwarder are disabled, the system DNS servers are assigned directly to the clients, but if that is not the case in practice for this setup, define them in the DHCP settings. If the client PC is not configured for DHCP, be sure it has the proper DNS servers set: either the LAN IP address of the firewall or an alternate set of working internal or external DNS servers.

Another possibility for DNS working from the firewall but not a local client is an overly strict firewall rule on the LAN. Check **Status > System Logs**, on the **Firewall** tab. If blocked connections appear in the log from the local client trying to reach a DNS server, then add a firewall rule at the top of the LAN rules for that interface which will allow connections to the DNS servers on **TCP and UDP** port **53**.

33.13 Troubleshooting the DNS Cache

33.13.1 DNS Forwarder

To clear the DNS Forwarder cache, restart the *dnsmasq* daemon as follows:

- Click **Status > Services**
- Find **dnsmasq** in the list
- Click  , or stop the service using  then start again with  .

Restarting the daemon will clear the internal cache, but the client PCs may still have cached entries.

33.13.2 Client DNS Cache

The DNS cache on a Windows PC may be cleaned from a command prompt or **Start > Run**:

```
ipconfig /flushdns
```

This may need to be executed from an Administrator command prompt on Windows Vista and later.

Other operating systems will surely have other means to clear the DNS resolver cache. For example, Ubuntu-based distributions also use *dnsmasq*, and it may be restarted using:

```
sudo service network-manager restart
```

Browsers also have their own internal DNS caches separate from the OS. Close and re-open the browser if none of the above help.

As a last resort, rebooting/restarting a client will surely clear any locally cached data.

33.14 Troubleshooting the DNS Forwarder

On rare occasions one might need to troubleshoot issues with certain queries to the DNS Forwarder (dnsmasq) or DNS Resolver (Unbound). In such cases it can be helpful to view the queries received by the DNS Forwarder and to see the responses generated.

For the DNS Forwarder, this can be done by adding the following keyword to the Advanced Options box on a new line:

```
log-queries
```

For the DNS Resolver, add this line:

```
log-queries: yes
```

When saved, the DNS Forwarder/Resolver will begin logging the received queries and their replies, along with some information about the result such as whether it was pulled from the cache.

Here are some examples of exchanges that might find in the query log:

A query using the system's defined DNS servers to answer the query (the answer was not yet known):

```
Dec  3 08:51:27 dnsmasq[1068]: query[A] daisy.ubuntu.com from 192.0.2.5
Dec  3 08:51:27 dnsmasq[1068]: forwarded daisy.ubuntu.com to 8.8.8.8
Dec  3 08:51:27 dnsmasq[1068]: forwarded daisy.ubuntu.com to 8.8.4.4
Dec  3 08:51:27 dnsmasq[1068]: reply daisy.ubuntu.com is 91.189.95.55
```

A query where the response was given from the DNS cache:

```
Dec  3 08:56:46 dnsmasq[1068]: query[A] dnl-14.geo.kaspersky.com from 10.0.10.128
Dec  3 08:56:46 dnsmasq[1068]: cached dnl-14.geo.kaspersky.com is 4.28.136.39
```

A cached negative response:

```
Dec  3 08:56:49 dnsmasq[1068]: query[A] wpad.example.com from 192.0.2.5
Dec  3 08:56:49 dnsmasq[1068]: cached wpad.example.com is NXDOMAIN-IPv4
```

A query where the reply cannot be sent because of an improper client IP (subnet ID, invalid IP):

```
Dec  3 08:49:21 dnsmasq[1068]: query[A] teredo.ipv6.microsoft.com from 192.0.2.0
Dec  3 08:49:21 dnsmasq[1068]: forwarded teredo.ipv6.microsoft.com to 8.8.8.8
Dec  3 08:49:21 dnsmasq[1068]: forwarded teredo.ipv6.microsoft.com to 8.8.4.4
Dec  3 08:49:21 dnsmasq[1068]: reply teredo.ipv6.microsoft.com.nsatc.net is 157.56.
↪144.215
Dec  3 08:49:21 dnsmasq[1068]: failed to send packet: Permission denied
```

33.15 Troubleshooting Disk and Filesystem Issues

33.15.1 Triggering a Filesystem Check

pfSense will run a filesystem check (`fsck`) at boot when it detects an unclean UFS filesystem, typically from after a power outage or other sudden unclean reboot or shutdown. In rare cases, that isn't always enough, as a filesystem can become corrupted in other ways that may not always leave the drive marked unclean.

Note: This is not necessary for ZFS. There is no `fsck` equivalent for ZFS, and it is not prone to the issues for which UFS and other filesystem types require checks and repairs.

The command `zpool scrub <pool name>` can validate the contents of a pool, identify potential issues, and attempt to repair damage where possible. The scrub operation is not the same as `fsck`; it is not necessary in cases where `fsck` is typically needed and it does not require a read-only filesystem so it can be run at any time.

In these cases, perform one of the following repair methods.

Automatic Filesystem Check

These methods force a filesystem check during the boot sequence even if the drive is considered clean.

Note: This option is not present on all firewalls as it is not compatible with certain hardware. To run a manual check instead, see *Manual Filesystem Check*.

GUI

- Navigate to **Diagnostics > Reboot**
- Set **Reboot Method** to *Reboot with Filesystem Check*
- Click  Submit

The firewall will reboot and run the check. Monitor the console output for errors.

Console

- Connect to the console
- Choose the menu option to reboot from the console menu (`5`)
- Enter `F` (uppercase "f")

The firewall will reboot and run the check. Monitor the console output for errors.

Manual Filesystem Check

If an automatic filesystem check is not possible, run a manual check instead:

- Reboot the firewall into single user mode.

  For most firewalls this can be done using option `2` from the boot menu.

  If the boot menu is unavailable, press the `Space` bar while the kernel is loading at boot time, which will start the boot loader with an `OK` or `loader>` prompt. From there, enter:

  ```
  boot -s
  ```

  Warning: Check the prompt again before typing this command. It must say either `loader>` or `OK`. If the prompt shows a different string such as `Marvel>>` that is not the correct prompt for this action. Reboot and try again.

- Press `Enter` when prompted for a shell
- Enter `fsck -fy /`
- Repeat the command at least **five** times, or until no errors are found nor fixed, even if the filesystem is reported clean.

Example:

```
/boot/kernel/kernel data=0x19e4818+0x777e8 syms=[0x4+0x9a3b0+0x4+0xdc388]
|
Hit [Enter] to boot immediately, or any other key for command prompt.


Type '?' for a list of commands, 'help' for more detailed help.
loader> boot -s
[lots of boot output]
Enter full pathname of shell or RETURN for /bin/sh:
# fsck -fy /
```

See also:

The Netgate Resource Library contains a video which walks through the process of running a filesystem check.

33.16 Troubleshooting Full Filesystem or Inode Errors

Error messages may appear on the console or main system log indicating that the hard drive in the firewall is full, such as:

```
/: write failed, filesystem is full
/: create/symlink failed, no inodes free
Warning: session_start(): open(/tmp/sess_XXXXXX, O_RDWR) failed: No space left on␣
↪device
```

The typical cause of such errors is rarely that the drive is full, but that the operating system is unable to contact the drive. In short, the disk (HDD, SSD, CF, etc) is dead or dying.

In cases when the drive is dying, the OS tries to write to the drive, and receives back an error code that there aren't any inodes left. Typically other messages about the underlying problem are logged to the console but not passed back to PHP so they can be seen by the GUI. Usually those refer to things like g_vfs_done or *Troubleshooting DMA and LBA Errors*.

In order to eliminate the possibility that the drive is actually full, such as from a package that went crazy eating up space, try to get to a shell from the console or ssh, and run the following command:

```
: df -hi
```

The output shows disk space usage for both capacity and inodes, using human-readable numbers. The **System Information** widget on the Dashboard also shows the usage for all mounted partitions. The following output is from a system running with UFS:

```
Filesystem                      Size    Used   Avail Capacity iused ifree %iused ␣
↪Mounted on
/dev/ufsid/5ec430415d66e7cc      29G    1.9G     25G     7%     25k  4.0M    1%   /
devfs                           1.0K    1.0K      0B   100%       0     0  100%   /dev
/dev/md0                        3.4M    120K    3.0M     4%      43   979    4%   /var/
↪run
devfs                           1.0K    1.0K      0B   100%       0     0  100%   /var/
↪dhcpd/dev
```

Note: The devfs lines do NOT indicate an actual problem; The devfs filesystem is virtual and used for housing device nodes not for files.

Of special concern is the space on `/var` and `/tmp` on systems configured to use RAM disks for those directories. Large files, such as an abnormally large DHCP leases file, can in fact fill up the `/var` memory disk and that is one way to encounter the problem.

If the root (`/`) slice has space and inodes remaining, and so do `/var` and `/tmp` and so on, then the problem is most likely a failing disk.

If disk space has been exhausted, find a way to free it up. This usually involves uninstalling or removing packages such as squid, or changing the settings so they use less space.

If the disk is failing, swap it out as soon as possible.

Often when the drive is failing or full the system will continue routing packets indefinitely until it would need to access the hard drive, at which point it would quit. Users have reported firewalls running for months with a dead drive unnoticed before, though of course that is not advisable.

Another possible cause is mild filesystem corruption, which could be helped by *Troubleshooting Disk and Filesystem Issues*.

33.17 Troubleshooting Thread Errors with Hostnames in Aliases

If a large number of hostnames are present in aliases, the following error may appear in the system log:

```
filterdns: Unable to create monitoring thread for host myhost.example.com! It
will not be monitored
```

This error indicated that the `filterdns` daemon hit the limit of the number of concurrent threads it could launch in order to resolve and monitor the hostnames properly.

The cure for this is to raise the number of allowed threads per process. This can be done by navigating from the pfSense® webGUI to **System > Advanced** on the **System Tunables** tab. There, create an entry for `kern.threads.max_threads_per_proc` and set it to `4096`.

After raising that value, `filterdns` must be restarted. This can be done by any method that triggers a filter reload, such as navigating to **Status > Filter Reload** and clicking **Reload Filter**.

33.18 Troubleshooting Firewall Rules

This section provides guidance for troubleshooting issues with firewall rules.

33.18.1 Check The Firewall Logs

The first step when troubleshooting suspected blocked traffic is to check the firewall logs (**Status > System Logs**, on the **Firewall** tab).

By default pfSense® will log all dropped traffic and will not log any passed traffic. Unless block or reject rules exist in the ruleset which do not use logging, all blocked traffic will be logged. If there are no log entries with a red  in the firewall logs which match the traffic in question, pfSense is not likely to be dropping the traffic.

33.18.2 Check the State Table

Attempt a connection and immediately check the state table at **Diagnostics > States** and filter on the source or destination to see if a state exists. If a state table entry is present, the firewall has passed the traffic.

If the rule in question is a pass rule, the state table entry means that the firewall passed the traffic through and the problem may be elsewhere and not on the firewall.

If the rule is a block rule and there is a state table entry, the open connection will **not** be cut off. To see an immediate effect from a new block rule, the states must be reset. See *Firewall States* for more information.

33.18.3 Review Rule Parameters

Edit the rule in question and review the parameters for each field. For TCP and UDP traffic, remember the source port is almost never the same as the destination port, and should usually be set to *any*.

If the default deny rule is to blame, craft a new pass rule that will match the traffic to be allowed. If the traffic is still blocked, there may be some other special aspect of the packets which require additional handling in the rule configuration. For example, certain multicast traffic may need to have **Allow IP Options** enabled, or the log entries may be due to asymmetric routing, or the packets may have an invalid combination of parameters such as a fragmented packet with "Don't Fragment" set inside.

See also:

See *Bypass Firewall Rules for Traffic on Same Interface* and *Static Route Filtering* for information on how to handle asymmetric routing.

In such advanced cases, running a packet capture for the traffic in question can help diagnose the problem. Refer to *Packet Capturing* for more information on how to capture and analyze packets.

Protocol

The protocol to which the rule will apply must be specified. Most often, this is TCP, UDP, or ICMP, but other protocols such as ESP, AH, and GRE are regularly encountered when dealing with VPNs.

Confusion arises when a firewall administrator is unsure of what protocol to use. A rule set with TCP may not work because the application being filtered may actually use UDP instead. When in doubt, try using TCP/UDP.

33.18.4 NAT Confusion

When crafting rules for firewalls involving inbound NAT connections, remember to use the **private IP address** as the Destination. This applies for port forwards as well as 1:1 NAT

33.18.5 Port Forward *pass* action

When creating a port forward, the *pass* action will bypass firewall rules and pass the traffic directly through without filtering. Change the setting to create an associated rule and then arrange the block rule above the resulting pass rule.

33.18.6 Source and Destination Ports

When crafting rules, bear in mind that typically only a source *or* a destination port needs to be specified, and rarely both. In the majority of cases, the source port does not matter at all. For example, to allow ssh access to the firewall, only specify a **destination** port of `22`. The source port of the client will be random.

33.18.7 Review Rule Ordering

Firewall rules are generally processed as follows:

- Floating Rules
- Interface Group rules
- Interface tab rules

See also:

See *Ordering of NAT and Firewall Processing* for more details.

If a floating rule with **quick** checked passed the traffic, then a block rule on an interface would have no chance to match the traffic.

33.18.8 Rules and Interfaces

Ensure rules are on the correct interface to function as intended. Traffic is filtered only by the ruleset configured on the interface where the traffic is *initiated*. Traffic coming from a system on the LAN destined for a system on any other interface is filtered by **only** the LAN rules. The same is true for all other interfaces.

33.18.9 Enable Rule Logging

Determine which rule is matching the traffic in question. The hit counters in the rule list can help with this to some degree. By enabling logging on pass rules, the firewall logs will show an individual entry specifically to determine which rule passed the connection.

33.18.10 Troubleshooting with packet captures

Packet captures can be invaluable for troubleshooting and debugging traffic issues. With a packet capture, it is easy to tell if the traffic is reaching the outside interface or leaving an inside interface, among many other uses. See *Packet Capturing* for more details on troubleshooting with packet captures.

33.18.11 New Rules Are Not Applied

If a new rule does not appear to apply, there are a couple possible explanations.

First, If the rule is a block rule and there is a state table entry, the open connection will **not** be cut off. See *Check the State Table*.

Second, the ruleset may not be reloading properly. Check **Status > Filter Reload** to see if an error is displayed. Click the  **Reload Filter** button on that page to force a new filter reload. If an error is displayed, resolve the problem as needed. If the cause is not obvious, consult support resources for assistance.

If none of the above causes are to blame, it's possible that the rule is not matching at all, so review the traffic and the rule again.

33.18.12 Unfilterable Traffic

Certain traffic cannot be filtered. Not because the pfSense software isn't capable, but because they actually do not touch the firewall at all. A prime example of this is trying to keep one device on the LAN from accessing another device on the same LAN. This is not possible if both clients are on the same subnet and switch; In that case, the routing of packets is handled at the switch level (layer 2), and the firewall has no knowledge of the traffic. If there is a need to control access in this way, the devices in question must be on separate firewall interfaces. When on different "legs" of the network, their traffic will route through the firewall, the firewall will have full control of the flow.

33.18.13 UPnP / NAT-PMP passed traffic

If *UPnP/NAT-PMP* is enabled and a LAN device opens a port to the world, the traffic may still get in even if it appears it should otherwise be blocked.

33.18.14 Asymmetric Routing

If reply traffic such as `TCP:A`, `TCP:SA`, or `TCP:RA` is shown as blocked in the logs, the problem could be asymmetric routing. See *Troubleshooting Asymmetric Routing* for more info.

33.18.15 Ruleset Failing to Load

It is also possible that the rules are not being loaded properly. Typically this would result in a notification in the GUI, however manual tests can be performed to check.

From the GUI, visit **Status > Filter Reload**. Click  **Reload Filter** wait for the process to stop, then scroll to the bottom of the page to see if the last line says `Done.` or if it stops. If it stops, for example in a particular package, then there may be a problem with that package.

The ruleset can also be verified from the console or **Diagnostics > Command** in the **Shell Execute** box by running:

```
pfctl -f /tmp/rules.debug
```

If an error is displayed, it may have an obvious fix, or search for that error to find possible resolutions.

33.18.16 Other Causes

There are other pitfalls in firewall rules, NAT, routing, and network design that can interfere with connectivity. See *Troubleshooting Network Connectivity* for more suggestions.

See also:

pfSense Hangouts on Youtube to view the June 2016 hangout on Connectivity Troubleshooting which contains much more detailed troubleshooting procedures.

33.19 Troubleshooting Bogon Network List Updates

Make sure the firewall can resolve DNS host names and can reach the bogons host, otherwise the update will fail.

To ensure the firewall can resolve the bogon update host via DNS, perform a *DNS Lookup*:

- Navigate to **Diagnostics > DNS Lookup**
- Enter `files.pfsense.org` in the **Hostname** field
- Click  **Lookup**

If that fails, *troubleshoot DNS resolution* for the firewall itself.

If that works, then perform a *port test* as demonstrated in Figure *Testing Connectivity for Bogon Updates*:

- Navigate to **Diagnostics > Test Port**

- Enter `files.pfsense.org` in the **Hostname** field
- Enter `80` in the **Port** field
- Click  **Test**

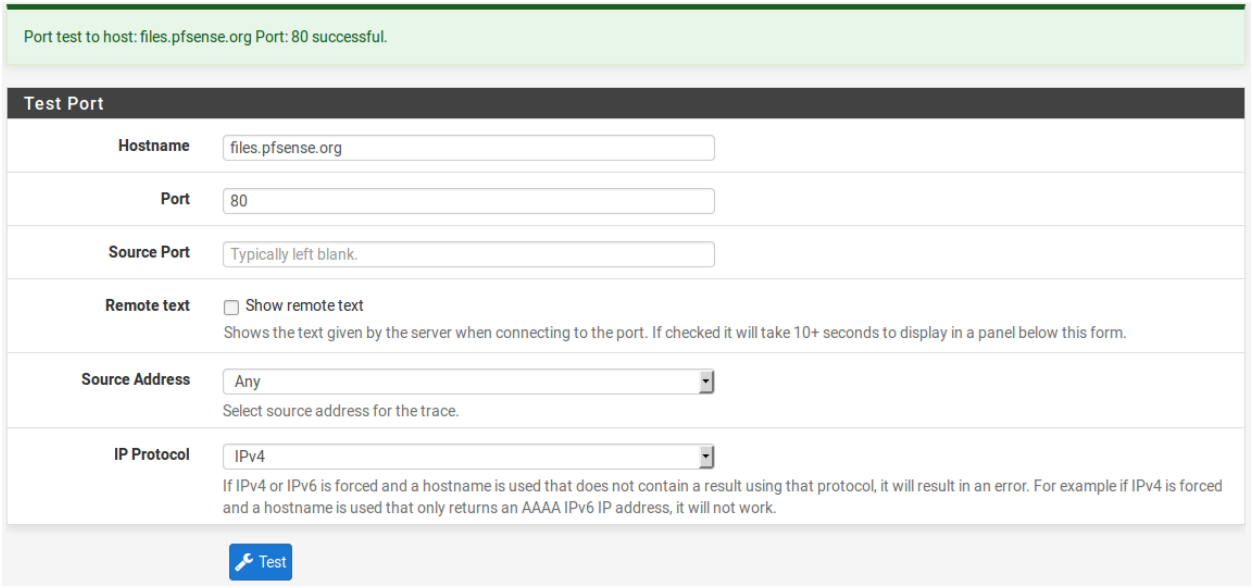


Fig. 2: Testing Connectivity for Bogon Updates

If that fails, *troubleshoot connectivity from the firewall*.

33.19.1 Forcing a Bogon Network List Update

With the relatively infrequent changes to the bogons list, and advance notice of new public IP assignments, a monthly bogons update is adequate. However there may be scenarios where a manual bogon update can help, such as if the bogon updates have been failing because of an incorrect DNS configuration. Execute an update via the GUI:

- Navigate to **Diagnostics > Tables**
- Select *bogons* or *bogonsv6* from the **Table** list
- Click  **Update**

33.20 Troubleshooting FTP Connections

In pfSense® software versions 2.0.x and 2.1.x, the FTP proxy is in -kernel. The only options to control its behavior are an on/off switch and a list of ports to be used by the proxy.

In pfSense software version 2.2.x and newer, there is no built-in FTP proxy. See *Using NAT and FTP without a Proxy*.

33.20.1 Disabling the FTP Proxy

If the FTP proxy must be disabled, this may be done by visiting **System > Advanced**, on the **System Tunables** tab, then set *debug.pfftpproxy=1*. Set it to *0* again to enable the proxy.

33.20.2 FTP Ports

FTP traffic is identified by the use of port 21. Other ports can be used if they added to a comma-separated list in the system tunable *debug.pfftpports* (e.g. *21,2121,4559*)

33.20.3 Rules to allow FTP

If problems are encountered with FTP, check the rules to/from FTP devices, ensure that both the control port and PASV range are allowed.

33.20.4 Troubleshooting/Alternatives

1. Disable the FTP Proxy and attempt the connection again
2. Use SCP/SFTP which only needs 1 port to traverse the firewall since it is wrapped in SSH (yes a safe AND simple way of traversing a firewall!)
3. Don't use FTP (highly recommended option)

33.21 Troubleshooting Gateway Monitoring

In some cases, the dpinger gateway monitoring daemon will output numeric error codes in the Gateways log indicating a problem reaching the monitored target IP address. The errors on this page are the most common.

33.21.1 sendto error: 55

```
55 ENOBUFS
No buffer space available.
An operation on a socket or pipe was not performed because the system lacked␣
↪sufficient buffer space or because a queue was full.
```

Several possible conditions can cause this. For a list of possible causes and solutions, see *No buffer space available*.

33.21.2 sendto error: 64

```
64 EHOSTDOWN
Host is down.
A socket operation failed because the destination host was down.
```

In this case, the firewall is unable to reach the a target host directly connected at layer 2 (No ARP response), or it received a similar error response from an upstream source. Generally this only happens due to remote problems, indicating that the target is actually down or the L1/L2 link to the target is down.

33.21.3 sendto error: 65

```
65 EHOSTUNREACH
No route to host.
A socket operation was attempted to an unreachable host.
```

Either there is no possible route to the target locally, or status information was received from an upstream router that indicated the same condition elsewhere along the path to the target.

This can happen due to a lack of default route, missing interface link route, or similar conditions.

33.22 Troubleshooting High Availability DHCP Failover

- The system time on both cluster nodes must be within 90 seconds of each other. Otherwise the time difference is too large and the DHCP daemon processes will not communicate.
- The interfaces must be assigned identically on both nodes, for example: wan=WAN, lan=LAN, opt1=Sync, opt2=DMZ. Check the `config.xml` contents directly to ensure a match.
- Look at the pool status section at **Status > DHCP leases**. All defined pools (often 1 per interface) are listed here. If any of the pools are in a state other than "normal", then debug the problem.
- Stop and restart the DHCP daemon from **Status > Services** on both nodes and check the status after a few moments
- Check the CARP VIP configuration for VIPs on interfaces used for DHCP failover. The primary node must have an **Advertising Frequency Skew** value *below* `20`, the secondary node must have an **Advertising Frequency Skew** value *above* `20`.
- Both nodes must be running the same version of pfSense® software. Update both nodes to the newest available stable release if they do not match. Older versions may have problems with various aspects of DHCP failover that have already been corrected.
- If all else fails, stop the DHCP daemon on both nodes, remove the DHCP lease database from `/var/dhcpd/var/db/dhcpd.leases`, then start the daemons again.

33.23 Troubleshooting the HAProxy Package

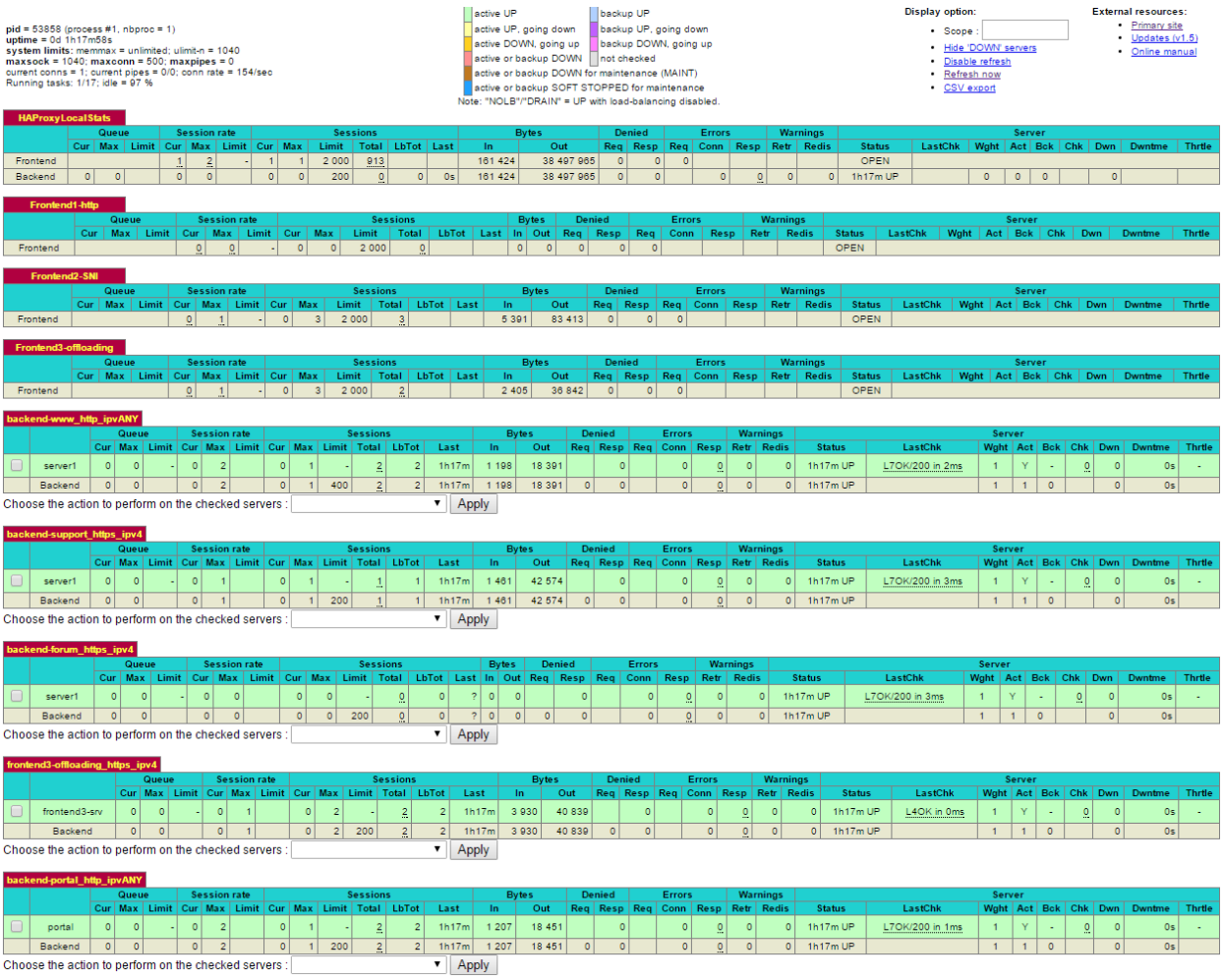
Troubleshooting steps for *HAProxy package*.

33.23.1 HAProxy Troubleshooting

For troubleshooting there are 2 parts are helpful, depending on the issue:

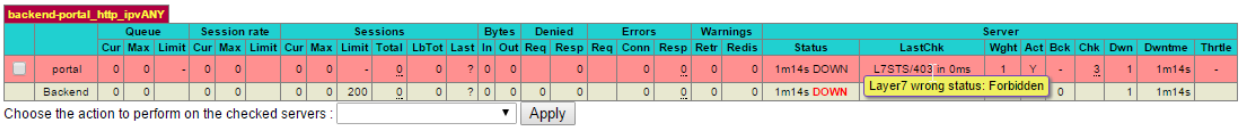
- Stats page
- Syslog logging

Stats

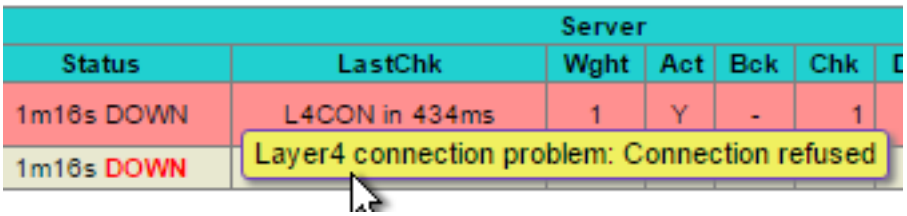


If health checks have been configured on the servers, the backend will show what servers are up or down. Layer 7 checks provide the most information about this, but a layer 6 or 4 check can also be useful.

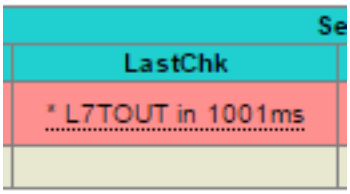
If a server is shown in red like here, hover over the check result for a second. It will tell what the short error code means in a more readable description:



There are different error codes that ask for different resolution.



or



A layer 4 issue might indicate that a wrong server ip or port was filled in, or that the server is not running / accepting connections. A firewall on the server itself, or a missing route could all cause these kinds of issues. A layer 6 issue indicates a problem with the SSL certificates. A layer 7 issue would generally be due to a unexpected or no status returned by the webserver, the webserver might take to long to present the checked url. Or the webserver does not support the configured check method/options.

For configuring the healthcheck, the following options should be accepted by most webserver, but are more difficult to filter out from normal webserver logs:

```
option httpchk GET / HTTP/1.1\r\nHost:\ www.yourdomain.com\r\nAccept:\ */*
```

If the backend requires authentication, another option could be to change the url to a different page that does not need authentication, perhaps specifically added to the webserver for this purpose:

```
option httpchk GET /healthcheck.php
```

Or accept that authentication error as the 'valid' result:

```
http-check expect status 401
```

Syslogs

If all backend servers are 'up' in the stats, but 'sometimes' users are reporting problems, then logging is important to configure and collect.

Haproxy allows for configuring syslog server destination on the settings tab. The actual logging to files must by configured on that destination syslog-server. The default log format is rather detailed if configured for the appropriate format. As such the 'Detailed logging' option in the frontend edit page should be checked. For mode HTTP servers the following will be configured in the config file:

```
option httplog
```

For mode TCP servers the following will be configured in the config file:

```
option tcplog
```

If needed, it's also possible to capture additional traffic headers which will be added into the syslog messages.

More information about these options can be found in the official documentation: Official HAProxy manual

33.24 Troubleshooting VPN Connectivity to a High Availability Secondary Node

If there is a VPN connection to a CARP cluster (site-to-site or mobile/road warrior), often one can communicate with the master but the backup node is unreachable.

The reason for this is that the VPN is configured on both firewalls. The packet from the client goes to the primary over the VPN tunnel that is UP, then goes from the primary to the secondary, and the secondary attempts to send it back over its own copy of the VPN which is DOWN because it is the backup. The packet never makes it back to the user.

To fix it, manual outbound NAT must be configured so that the firewall does NAT on the traffic from the VPN subnet going to the secondary node's IP, and vice versa. that way it appears to originate from the opposing firewall and not the VPN, so the traffic returns as expected. Manual Outbound NAT is likely already set since is is typically a requirement of CARP. If not, Manual Outbound NAT must be enabled for this to work, or Hybrid Outbound NAT may be used on pfSense® software version 2.2 and later.

For example, add a manual outbound NAT rule on the LAN interface, source being the VPN subnet, destination being an alias that contains both the primary and secondary node LAN IPs. Translation would be **Interface Address** (NOT the CARP VIP!).

With the NAT rule present when attempting to access the opposing node over the VPN the traffic will appear to originate from the node to which the VPN is currently connected and the return traffic will go back as expected.

33.25 Troubleshooting High Availability

High availability configurations can be complex, and with so many different ways to configure a failover cluster, it can be tricky to get things working properly. In this section, some common (and not so common) problems will be discussed and hopefully solved for the majority of cases. If issues are still present after consulting this section, there is a dedicated HA/CARP/VIPs board on the Netgate Forum.

Before proceeding, take the time to check all members of the HA cluster to ensure that they have consistent configurations. Often, it helps to walk through the example setup, double checking all of the proper settings. Repeat the process on the secondary node, and watch for any places where the configuration must be different on the secondary. Be sure to check the CARP status (*Check CARP status*) and ensure CARP is enabled on all cluster members.

Errors relating to HA will be logged in **Status > System Logs**, on the **System** tab. Check those logs on each system involved to see if there are any messages relating to XMLRPC sync, CARP state transitions, or other related errors.

See also:

The issues on this page are for HA in general. For issues specific to using HA in virtual environments, see *Troubleshooting High Availability Clusters in Virtual Environments*

33.25.1 Common Misconfigurations

There are several common misconfigurations that happen which prevent HA from working properly.

Incorrect Interface Order

As mentioned on *pfSense XML-RPC Config Sync Overview*, the interface assignment order and internal identifiers must match identically on both nodes.

If the interface order does not match, the configuration synchronziation process will copy rules and other settings such as DHCP failover to the wrong interfaces on the secondary node.

Use a different VHID on each CARP VIP

A different VHID must be used on each CARP VIP created on a given interface or broadcast domain. The VHID determines the virtual MAC address used by that CARP IP address, this different clusters attempting to use the same VHID on the same L2 segment cause a MAC address conflict.

With a single HA pair, input validation will prevent duplicate VHIDs. Unfortunately it isn't always that simple. CARP is a multicast technology, and as such anything using CARP on the same network segment must use a unique VHID. VRRP also uses a similar protocol as CARP, so ensure there are no conflicts with VRRP VHIDs, such as if the ISP or another router on the local network is using VRRP.

The best way around this is to use a unique set of VHIDs. If a known-safe private network is in use, start numbering at 1. On a network where VRRP or CARP are conflicting, consult with the administrator of that network to find a free block of VHIDs.

Incorrect CARP VIP Settings

Inspect the settings for CARP VIPs (**Firewall > Virtual IPs**) to ensure they are correct and consistent on both nodes.

The **Advertising Frequency** values must be appropriate for each VIP and node:

Base Values should be the same on both nodes. In some situations where the secondary node is on a slow or non-local link, users have increased this value on only the secondary, but that can lead to problems with each node assuming their expected roles at the proper times.

Skew Values must be different on the primary and secondary nodes. The primary is typically `1` or `0`, and the secondary is typically `100`.

Incorrect Times

Check that all systems involved are properly synchronizing their clocks and have valid time zones, especially if running in a Virtual Machine. If the clocks are too far apart, some synchronization tasks like DHCP failover will not work properly.

Incorrect Subnet Mask

The real subnet mask must be used for a CARP VIP, not /32. This must match the subnet mask for the IP address on the interface to which the CARP IP is assigned.

Both Nodes in Maintenance Mode

If both nodes have activated **Persistent CARP Maintenance Mode** at **Status > CARP (failover)**, they each will advertise a skew of `254` and the actual status will be unpredictable. Ensure only one node is in maintenance mode at a time.

33.25.2 Incorrect Hash Error

There are a few reasons why this error turns up in the system logs, some more worrisome than others.

If CARP is not working properly when this error is present, it could be due to a configuration mismatch. Ensure that for a given VIP, that the VHID, password, and IP address/subnet mask all match.

If the settings appear to be proper and CARP still does not work while generating this error message, then there may be multiple CARP instances on the same broadcast domain. Disable CARP and monitor the network with tcpdump (*Packet Capturing*) to check for other CARP or CARP-like traffic, and adjust VHIDs appropriately.

If CARP is working properly, and this message is in the logs when the system boots up, it may be disregarded. It is normal for this message to be seen when booting, as long as CARP continues to function properly (primary shows **MASTER**, secondary shows **BACKUP** for status).

33.25.3 Both Systems Appear as MASTER

This will happen if the secondary cannot see the CARP advertisements from the primary. Check for firewall rules, connectivity trouble, switch configurations. Also check the system logs for any relevant errors that may lead to a solution. If this is encountered in a Virtual Machine (VM) Product such as ESX, see *Troubleshooting High Availability Clusters in Virtual Environments*.

33.25.4 Primary system is stuck as BACKUP

In some cases, this is may happen normally for a short period after a system comes back online. However, certain hardware failures or other error conditions can cause a server to silently take on a high advskew of 240 in order to signal that it still has a problem and should not become master. This can check be checked from the GUI, or via the shell or **Diagnostics > Command**.

In the GUI, this condition is printed in an error message on **Status > CARP**.

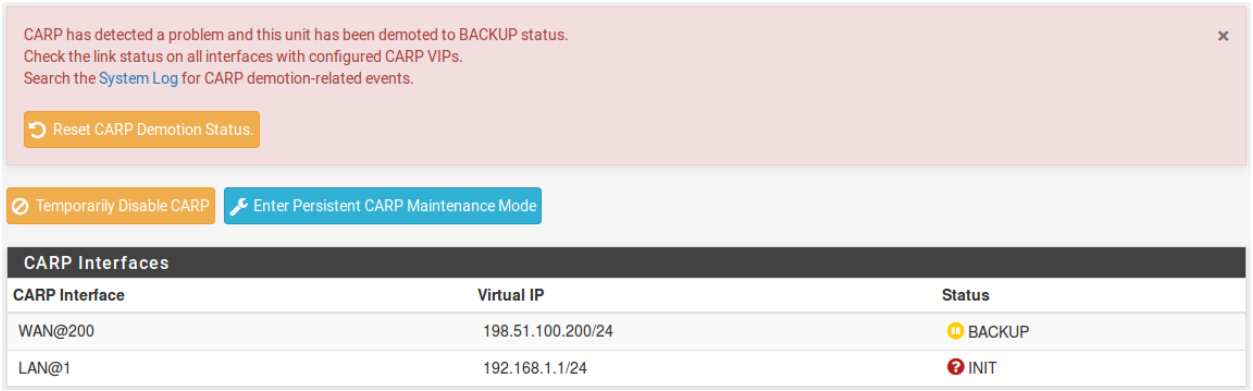


Fig. 3: CARP Status when Primary is demoted

From the shell or **Diagnostics > Command**, run the following command to check for a demotion:

```
#  sysctl net.inet.carp.demotion
net.inet.carp.demotion: 240
```

If the value is greater than `0`, the node has demoted itself.

In that case, isolate the firewall, check its network connections, and perform further hardware testing.

If the demotion value is 0 and the primary node still appears to be demoting itself to BACKUP or is flapping, check the network to ensure there are no layer 2 loops. If the firewall receives back its own heartbeats from the switch, it can also trigger a change to BACKUP status.

33.25.5 Other Switch and Layer 2 Issues

- If the units are plugged into separate switches, ensure that the switches are properly trunking and passing broadcast/multicast traffic.
- Some switches have broadcast/multicast filtering, limiting, or "storm control" features that can break CARP.
- Some switches have broken firmware that can cause features like IGMP Snooping to interfere with CARP.
- If a switch on the back of a modem/CPE is use, try a real switch instead. These built-in switches often do not properly handle CARP traffic. Often plugging the firewalls into a proper switch and then uplinking to the CPE will eliminate problems.

33.25.6 Configuration Synchronization Problems

Double check the following items when problems with configuration synchronization are encountered:

- The username must be **admin** on all nodes.
- The password in the configuration synchronization settings on the primary must match the password on the backup.
- The WebGUI must be on the same port on all nodes.
- The WebGUI must be using the same protocol (HTTP or HTTPS) on all nodes.
- Traffic must be permitted to the WebGUI port on the interface which handles the synchronization traffic.
- The pfsync interface must be enabled and configured on all nodes.
- Verify that only the primary sync node has the configuration synchronization options enabled.
- Ensure no IP address is specified in the **Synchronize Config to IP** on the secondary node.
- Ensure the clocks on both nodes are current and are reasonably accurate.

33.25.7 HA and Multi-WAN Troubleshooting

If trouble is encountered reaching CARP VIPs from when dealing with Multi-WAN, double check that a rule is present like the one mentioned in *Firewall Configuration*

33.26 Troubleshooting High Availability Clusters in Virtual Environments

33.26.1 Hypervisor users (Especially VMware ESX/ESXi)

The below settings are specifically for VMware ESX/ESXi but similar settings may be present on Hyper-V, VirtualBox, and other similar hypervisors.

1. Enable promiscuous mode on the vSwitch
2. Enable **MAC Address changes**
3. Enable **Forged transmits**
4. If multiple physical ports exist on the same vswitch, the Net.ReversePathFwdCheckPromisc option must be enabled to work around a vswitch bug where multicast traffic will loop back to the host, causing CARP to not function with "link states coalesced" messages. (See below)

ESX VDS Promisc Workaround

If a Virtual Distributed Switch is in use, a port group can be made for the firewall interfaces with promiscuous mode enabled, and a separate non-promiscuous port group may be used for other hosts. This has been reported to work by users on the forum as a way to strike a balance between the requirements for letting CARP function and for securing client ports.

ESX VDS Upgrade Issue

If a VDS (Virtual Distributed Switches) is used in ESX 4.0 or 4.1 and an **upgrade** from 4.0 to 4.1 or 5.0 is performed, the VDS will not properly pass CARP traffic. If a new VDS is **created** on 4.1 or 5.0, it will work, but the upgraded VDS will not.

It is reported that disabling promiscuous mode on the VDS and then re-enabling it will resolve the issue.

ESX VDS Port Mirroring Issue

If port mirroring is enabled on a VDS, it will break promiscuous mode. To fix it, disable promiscuous mode, then re-enable promiscuous mode.

Client Port Issues

If a physical CARP cluster is connected to a switch with an ESX box using multiple ports on the ESX box (lagg group or similar), and only certain devices/IPs are reachable by the target VM, then the port group settings in ESX may need adjusted to set the load balancing for the group to hash based on IP, not the originating interface.

Side effects of having that set incorrectly include:

- Traffic only reaching the target VM in promisc mode on its NIC
- Inability to reach the CARP IP from the target VM when the "real" IP of the primary firewall is reachable
- Port forwards or other inbound connections to the target VM work from some IPs and not others.

Changing Net.ReversePathFwdCheckPromisc

Login VMware vSphere Client

For each VMware host

- Click on host to configure and select **Configuration** Tab
- Click **Software Advanced Settings** in left pane
- Click on **Net** and scroll down to **Net.ReversePathFwdCheckPromisc** and set to *1*
- Click **OK**

Promiscuous Mode interfaces need to be set now or twiddled off and then back on. This is done per host by clicking Networking in the Hardware section

- For each vSwitch and/or Virtual Machine Port Group.
  - NOTE: If Promiscuous is already enabled it must be disabled, saved and then re-enabled, saved.
  - Click on **Properties** of vSwtich
  - By Default Promiscuous Mode is **Reject**.
  - To Change click **Edit > Security** Tab
  - Select **Accept** from drop down
  - Click **OK**.
- However, this setting is usually applied per Virtual Machine Port Group (More Secure) where the VSwitch is left at default to Reject.
  - **Edit > Security > Policy Exceptions**
  - Uncheck Promiscuous Mode
  - Click **OK**
  - **Edit > Security > Policy Exceptions**
  - Check Promiscuous Mode and select **Accept**.

More information available from VMware

ESX Physical NIC Failure Fails to Trigger Failover

Self-demotion in CARP relies on the loss of link on a switch port. As such, if a primary and secondary firewall instance are on separate ESX units and the primary unit loses a switch port link and does not expose that to the VM, CARP will stay MASTER on all of its VIPs there and the secondary will also believe it should be MASTER. One way around this is to script an event in ESX that will take down the switch port on the VM if the physical port loses link. There may be other ways around this in ESX as well.

VMware Workstation

If using VMware workstation on Linux for testing/modeling and CARP does not function, it is likely because VMware workstation is running non-root and cannot set the vmnet adapter in Promiscuous mode.

The permissions on `/dev/vmnet*` should be changed such that the user running VMware workstation is allowed to modify the `/dev/vmnet*` devices. See the VMware KB for details.

To make the change permanent, edit **/etc/init.d/vmware**, and in function **vmwareStartVmnet()**, add commands to **chgrp** and **chown** the vmnet devices to a group which contains user running VMware Workstation.

33.26.2 KVM+QEMU Issues

Be sure to use e1000 NICs (em(4)), not the ed(4) NICs or CARP VIPs will never leave init state.

33.26.3 VirtualBox Issues

From this thread:

- Setting **Promiscuous mode: Allow All** on the relevant interfaces of the VM allows CARP to function on any interface type (Bridged, Host-Only, Internal)

33.27 Troubleshooting High CPU Load

First, open a shell from SSH or the serial/VGA console (option `8`).

Typically one of these commands will include some obvious consumer of large amounts of system resources. For example, if the system CPU usage is high, it may be the packet filter. If a VPN process is using lots of CPU, the hardware may not be able to process more VPN traffic. If it is a NIC, see *Hardware Tuning and Troubleshooting*, etc.

If the solution is not obvious based on the output, post the collected information on the forum or contact support for further assistance.

33.27.1 View CPU Processes

To view the top processes, including interrupt processing CPU usage and system CPU:

```
top -aSH
```

33.27.2 View Interrupt Counters

To view the interrupt counters and other system usage:

```
systat -vmstat 1
```

33.27.3 View mbuf Usage

To view the mbuf usage:

```
netstat -m
```

Note: Alternately, check the dashboard mbuf counter, and the graph under **Status > Monitoring** on the **System** tab.

33.27.4 View I/O Operations

To view I/O operations:

```
systat -iostat 1
```

Or:

```
top -aSH
```

Then press `m` to switch to **I/O mode** to view disk activity.

33.28 Troubleshooting Installation Issues

The vast majority of the time, installations of pfSense® software finish with no problems. The following sections describe the most common problems and the steps to resolve them.

See also:

Troubleshooting Boot Issues

33.28.1 Boot from Install Media Fails

Due to the wide array of hardware combinations in use, it is not uncommon for a memstick or CD to boot incorrectly (or not at all). Given the unpredictable nature of commodity hardware support, using hardware from the Netgate Store is the only guaranteed path to success.

That said, the most common problems and solutions are:

USB Memstick Support Some BIOS implementations can be picky about USB memstick support. If booting from one stick fails, try a different one.

USB 3 Ports Certain combinations of USB sticks and ports, especially USB 3.0 ports, may not work correctly. Try a USB 2.0 memstick in a USB 2.0 port.

BIOS Issues Update to the most recent BIOS, and disable any unneeded peripherals such as Firewire, Floppy Drives, and Audio.

Dirty Optical Drive Clean the drive with a cleaning disc or a can of compressed air, or try another drive.

Bad Optical Media Burn another disc and/or burn the disc at a lower speed. Perhaps try another brand of media.

SATA/IDE Cable Issues Try a different SATA/IDE cable between the CD-ROM drive and the controller or motherboard

Boot Loader Issues There have been cases where specific versions of the CD boot loader from FreeBSD did not work on certain hardware. In these cases, see *Alternate Installation Techniques* to perform the target drive installation on a separate PC and then move it to the target hardware.

33.28.2 Boot from hard drive after installation fails

After the installation completes and the firewall restarts, there are conditions which may prevent the operating system from fully booting. The most common reasons are typically BIOS-related. For example, a BIOS implementation may not boot from a disk using GPT or ZFS, or may require UEFI.

Some of these may be worked around by choosing different options for the partition layout during the installation process. Upgrading the BIOS to the latest version available may also help.

Altering the SATA options in the BIOS has improved booting in some situations as well. If a SATA hard drive is being used, experiment with changing the SATA options in the BIOS for settings such as AHCI, Legacy, or IDE. AHCI is the best mode to use with current versions of pfSense software.

See also:

For additional troubleshooting techniques, see *Troubleshooting Boot Issues*.

33.28.3 Interface link up not detected

If the firewall complains that it did not detect an interface link up event during automatic assignment, first make sure that the cable is unplugged and that the interface does not have a link light prior to choosing the link detection option. After selecting the option, plug the cable back into the interface and ensure it has a link light prior to pressing `Enter`. Test or replace the cable in question if it does not show a link light on the switch and/or NIC port once it is connected.

If a network cable is connected directly between two computers and not to a switch, and one of those pieces of hardware is older (e.g. 10/100 NIC) ensure that a crossover cable is being used. Gigabit adapters all support Auto-MDIX and will handle this internally, but many older 10/100 adapters do not. Similarly, if connecting a firewall running pfSense software to a switch that does not support Auto-MDIX, use a straight-through patch cable.

If the interface is properly connected but the firewall still does not detect the link event, the network interface may not properly detect or report link status to the operating system or driver. In this case, manually assigning the interfaces is necessary.

33.28.4 Hardware Troubleshooting

the following suggestions will help resolve general hardware issues.

Booting from USB

If the boot stops with a `mountroot>` prompt while booting off the installer image, usually with USB CD/DVD drives, escape to the loader prompt from the boot menu and run the following:

```
set kern.cam.boot_delay=10000
boot
```

At which point the boot will continue normally.

If the firewall is running permanently from a medium that requires this delay, edit `/boot/loader.conf.local` and insert the following line:

```
kern.cam.boot_delay=10000
```

Remove unnecessary hardware

If the firewall contains hardware that will not be used, remove or disable it. This normally isn't an issue, but can cause problems and has the potential to reduce performance. If an unused piece of hardware is removable, take it out of the firewall or disable it in the BIOS.

Disable PNP OS in the BIOS

This is a common fix for older hardware. BIOS configuration screens may contain a setting for **PNP OS** or **Plug and Play OS**, which should be set to `disable` or `no` . A few have a setting for OS, which should usually be set to `other`.

Upgrade the BIOS

The second most common fix for hardware problems is upgrading the BIOS to the latest revision. People seem to have a hard time believing this one, but trust us, do it. BIOS updates commonly fix bugs in hardware. It isn't uncommon to hit problems induced by hardware bugs on systems that have been stable running Windows for long periods of time. Either Windows doesn't trigger the bug, or has a work around, as we have encountered this on multiple occasions. Things that BIOS updates can fix include: Failing to boot, time keeping problems, general instability, and other issues like hardware compatibility.

Reset BIOS settings to factory defaults

Recycled systems may have an atypical BIOS configuration. Most contain an option allowing factory default options to be loaded. Use this option to get a fresh start on the BIOS settings.

Other BIOS settings

If the BIOS allows power management configuration, try toggling that option. Look for anything else that seems relevant to whichever aspect of the installation is failing. If it gets to this point, the target hardware is probably a lost cause and alternate hardware may be necessary. Also check to see if the BIOS has an event log that may list hardware errors such as memory test failures.

If the hardware uses a new or recent chipset, a development version of pfSense software may work. Check the Snapshots page to see if there is a development (e.g. Beta or release candidate) build to try.

Other Hardware Issues

The target hardware may be faulty, which testing with diagnostic software may reveal. Test the hard drive with diagnostic software from the OEM, and test the memory with a program such as memtest86+. These and more tools are available on the "Ultimate Boot CD", which is preloaded with many free hardware diagnostic tools.

Also ensure that all of the fans are spinning at speed, and that no components are overheating. If this is older reused hardware, compressed/canned air cleaning of the fans and heat sinks can work wonders.

33.29 Troubleshooting IPsec VPNs

Due to the finicky nature of IPsec, it isn't unusual for trouble to arise. Thankfully there are some basic (and some not so basic) troubleshooting steps that can be employed to track down potential problems.

See also:

Troubleshooting Duplicate IPsec SA Entries

33.29.1 IPsec Logging

Examples presented in this chapter have logs edited for brevity but significant messages remain.

Logging for IPsec may be configured to provide more useful information. To configure IPsec logging for diagnosing tunnel issues with pfSense®, the following procedure yields the best balance of information:

- Navigate to **VPN > IPsec** on the **Advanced Settings** tab
- Set **IKE SA**, **IKE Child SA**, and **Configuration Backend** to *Diag*
- Set all other log settings to *Control*
- Click Save

Note: Changing logging options is not disruptive to IPsec activity and there is no need to enter a specific "debug mode" for IPsec on current versions of pfSense.

33.29.2 Tunnel does not establish

First check the service status at **Status > Services**. If the IPsec service is stopped, double check that it is enabled at **VPN > IPsec**. Also, if using mobile clients, ensure that on the **Mobile clients** tab, the enable box is also checked.

If the service is running, check the firewall logs (**Status > System Logs**, **Firewall** tab) to see if the connection is being blocked, and if so, add a rule to allow the blocked traffic. Rules are normally added automatically for IPsec, but that feature can be disabled.

The single most common cause of failed IPsec tunnel connections is a configuration mismatch. Often it is something small, such as a DH group set to 1 on side A and 2 on side B, or perhaps a subnet mask of /24 on one side and /32 on the other. Some routers (Linksys, for one) also like to hide certain options behind "Advanced" buttons or make assumptions. A lot of trial and error may be involved, and a lot of log reading, but ensuring that both sides match precisely will help the most.

Depending on the Internet connections on either end of the tunnel, it is also possible that a router involved on one side or the other does not properly handle IPsec traffic. This is a larger concern with mobile clients, and networks where NAT is involved outside of the actual IPsec endpoints. The problems are generally with the ESP protocol and problems with it being blocked or mishandled along the way. NAT Traversal (NAT- T) encapsulates ESP in UDP port 4500 traffic to work around these issues.

33.29.3 Tunnel establishes but no traffic passes

The top suspect if a tunnel comes up but won't pass traffic is the IPsec firewall rules. If Site A cannot reach Site B, check the Site B firewall log and rules. Conversely, if Site B cannot contact Site A, check the Site A firewall log and rules. Before looking at the rules, inspect the firewall logs at **Status > System Logs**, on the **Firewall** tab. If blocked entries are present which involve the subnets used in the IPsec tunnel, then move on to checking the rules. If there are no log entries indicating blocked packets, revisit the section on IPsec routing considerations in *Client Routing and Gateway Considerations*.

Blocked packets on the IPsec or *enc0* interface indicate that the tunnel itself has established but traffic is being blocked by firewall rules. Blocked packets on the LAN or other internal interface may indicate that an additional rule may be needed on that interface ruleset to allow traffic from the internal subnet out to the remote end of the IPsec tunnel. Blocked packets on WAN or OPT WAN interfaces would prevent a tunnel from establishing. Typically this only happens when the automatic VPN rules are disabled. Adding a rule to allow the ESP protocol and UDP port 500 from that remote IP address will allow the tunnel to establish. In the case of mobile tunnels, allow traffic from any source to connect to those ports.

Rules for the IPsec interface can be found under **Firewall > Rules**, on the **IPsec** tab. Common mistakes include setting a rule to only allow *TCP* traffic, which means things like ICMP ping and DNS would not work across the tunnel. See *Firewall* for more information on how to properly create and troubleshoot firewall rules.

In some cases it is possible that a setting mismatch can also cause traffic to fail passing the tunnel. In one instance, a subnet defined on one non-pfSense firewall was `192.0.2.1/24`, and on the pfSense firewall it was `192.0.2.0/24`. The tunnel established, but traffic would not pass until the subnet was corrected.

Routing issues are another possibility. Running a traceroute (tracert on Windows) to an IP address on the opposite side of the tunnel can help track down these types of problems. Repeat the test from both sides of the tunnel. Check the *Client Routing and Gateway Considerations* section in this chapter for more information. When using traceroute , traffic which enters and leaves the IPsec tunnel will seem to be missing some interim hops. This is normal, and part of how IPsec works. Traffic which does not properly enter an IPsec tunnel will appear to leave the WAN interface and route outward across the Internet, which would point to either a routing issue such as pfSense not being the gateway (as in *Client Routing and Gateway Considerations*), an incorrectly specified remote subnet on the tunnel definition, or to a tunnel which has been disabled.

33.29.4 Some hosts work, but not all

If traffic between some hosts over the VPN functions properly, but some hosts do not, this is commonly one of four things:

Missing, incorrect or ignored default gateway If the device does not have a default gateway, or has one pointing to something other than the pfSense firewall, it does not know how to properly get back to the remote network on the VPN (see *Client Routing and Gateway Considerations*). Some devices, even with a default gateway specified, do not use that gateway. This has been seen on various embedded devices, including IP cameras and some printers. There isn't anything that can be done about that other than getting the software on the device fixed. This can be verified by running a packet capture on the inside interface of the firewall connected to the network containing the device. Troubleshooting with tcpdump is covered in *Examples of using tcpdump on the command line*, and an IPsec-specific example can be found in *IPsec tunnel will not connect*. If traffic is observed leaving the inside interface of the firewall, but no replies return, the device is not properly routing its reply traffic or could potentially be blocking it via a local client firewall.

Incorrect subnet mask If the subnet in use on one end is `10.0.0.0/24` and the other is `10.254.0.0/24`, and a host has an incorrect subnet mask of `255.0.0.0` or `/8`, it will never be able to communicate across the VPN because it thinks the remote VPN subnet is part of the local network and hence routing will not function properly. The system with the broken configuration will attempt to contact the remote system via ARP instead of using the gateway.

Host firewall If there is a firewall on the target host, it may not be allowing the connections. Check for things like Windows Firewall, iptables, or similar utilities that may be preventing the traffic from being processed by the host.

Firewall rules on pfSense Ensure the rules on both ends allow the desired network traffic.

33.29.5 Connection Hangs

IPsec does not gracefully handle fragmented packets. Many of these issues have been resolved over the years, but there may be some lingering problems. If hangs or packet loss are seen only when using specific protocols (SMB, RDP, etc.), MSS clamping for the VPN may be necessary. MSS clamping can be activated under **VPN > IPsec** on the **Advanced Settings** tab. On that screen, check **Enable MSS clamping on VPN traffic** and then enter a value. A good starting point would be 1400, and if that works slowly increase the MSS value until the breaking point is hit, then back off a little from there.

33.29.6 "Random" Tunnel Disconnects/DPD Failures on Low-End Routers

If IPsec tunnels are dropped on low-end hardware that is pushing the limits of its CPU, DPD on the tunnel may need disabled. Such failures tend to correlate with times of high bandwidth usage. This happens when the CPU on a low-power system is tied up with sending IPsec traffic or is otherwise occupied. Due to the CPU overload it may not take the time to respond to DPD requests or see a response to a request of its own. As a consequence, the tunnel will fail a DPD check and be disconnected. This is a clear sign that the hardware is being driven beyond its capacity. If this happens, consider replacing the firewall with a more powerful model.

33.29.7 Tunnels Establish and Work but Fail to Renegotiate

In some cases a tunnel will function properly but once the phase 1 or phase 2 lifetime expires the tunnel will fail to renegotiate properly. This can manifest itself in a few different ways, each with a different resolution.

DPD Unsupported, One Side Drops but the Other Remains

Consider this scenario, which DPD is designed to prevent, but can happen in places where DPD is unsupported:

- A tunnel is established from Site A to Site B, from traffic initiated at Site A.
- Site B expires the phase 1 or phase 2 before Site A
- Site A will believe the tunnel is up and continue to send traffic as though the tunnel is working properly.
- Only when Site A's phase 1 or phase 2 lifetime expires will it renegotiate as expected.

In this scenario, the two likely things resolutions are: Enable DPD, or Site B must send traffic to Site A which will cause the entire tunnel to renegotiate. The easiest way to make this happen is to enable a keep alive mechanism on both sides of the tunnel.

33.29.8 Tunnel Establishes When Initiating, but not When Responding

If a tunnel will establish sometimes, but not always, generally there is a mismatch on one side. The tunnel may still establish because if the settings presented by one side are more secure, the other may accept them, but not the other way around. For example if there is an Aggressive/Main mode mismatch on an IKEv1 tunnel and the side set for *Main* initiates, the tunnel will still establish. However, if the side set to *Aggressive* attempts to initiate the tunnel it will fail.

Lifetime mismatches do not cause a failure in Phase 1 or Phase 2.

To track down these failures, configure the logs as shown in *IPsec Logging* and attempt to initiate the tunnel from each side, then check the logs.

33.29.9 IPsec Log Interpretation

The IPsec logs available at **Status > System Logs**, on the **IPsec** tab contain a record of the tunnel connection process and some messages from ongoing tunnel maintenance activity. Some typical log entries are listed in this section, both good and bad. The main things to look for are key phrases that indicate which part of a connection worked. If "IKE_SA ... established" is present in the log, that means phase 1 was completed successfully and a Security Association was negotiated. If "CHILD_SA ... established" is present, then phase 2 has also been completed and the tunnel is up.

In the following examples, the logs have been configured as listen in *IPsec Logging* and irrelevant messages may be omitted. Bear in mind that these are samples and the specific ID numbers, IP addresses, and so forth will vary.

Successful Connections

When a tunnel has been successfully established both sides will indicate that an **IKE SA** and a **Child SA** have been established. When multiple Phase 2 definitions are present with IKEv1, a child SA is negotiated for each Phase 2 entry.

Log output from the initiator:

```
charon: 09[IKE] IKE_SA con2000[11] established between 192.0.2.90[192.0.2.90]...192.0.
↪2.74[192.0.2.74]
charon: 09[IKE] CHILD_SA con2000{2} established with SPIs cf4973bf_i c1cbfdf2_o and␣
↪TS 192.168.48.0/24|/0 === 10.42.42.0/24|/0
```

Log output from the responder:

```
charon: 03[IKE] IKE_SA con1000[19] established between 192.0.2.74[192.0.2.74]...192.0.
↪2.90[192.0.2.90]
charon: 16[IKE] CHILD_SA con1000{1} established with SPIs c1cbfdf2_i cf4973bf_o and␣
↪TS 10.42.42.0/24|/0 === 192.168.48.0/24|/0
```

Failed Connection Examples

These examples show failed connections for varying reasons. In most cases it's clear from the examples that the initiator does not receive messages about specific items that do not match, so the responder logs are much more informative. This is done to protect the security of the tunnel, it would be insecure to provide messages to a potential attacker that would give them information about how the tunnel is configured.

Phase 1 Main / Aggressive Mismatch

In this example, the initiator is set for Aggressive mode while the responder is set for Main mode.

Log output from the initiator:

```
charon: 15[IKE] initiating Aggressive Mode IKE_SA con2000[1] to 203.0.113.5
charon: 15[IKE] received AUTHENTICATION_FAILED error notify
charon: 13[ENC] parsed INFORMATIONAL_V1 request 1215317906 [ N(AUTH_FAILED) ]
charon: 13[IKE] received AUTHENTICATION_FAILED error notify
```

Log output from the responder:

```
charon: 13[IKE] Aggressive Mode PSK disabled for security reasons
charon: 13[ENC] generating INFORMATIONAL_V1 request 2940146627 [ N(AUTH_FAILED) ]
```

Note that the logs on the responder state clearly that Aggressive mode is disabled, which is a good clue that the mode is mismatched.

In the reverse case, if the side set for Main mode initiates, the tunnel to a pfSense firewall will establish since Main mode is more secure.

Phase 1 Identifier Mismatch

When the identifier does not match, the initiator only shows that the authentication failed, but does not give a reason. The responder states that it is unable to locate a peer, which indicates that it could not find a matching Phase 1, which implies that no matching identifier could be located.

Log output from the initiator:

```
charon: 10[ENC] parsed INFORMATIONAL_V1 request 4216246776 [ HASH N(AUTH_FAILED) ]
charon: 10[IKE] received AUTHENTICATION_FAILED error notify
```

Log output from the responder:

```
charon: 12[CFG] looking for pre-shared key peer configs matching 203.0.113.5...198.51.
↪100.3[someid]
charon: 12[IKE] no peer config found
charon: 12[ENC] generating INFORMATIONAL_V1 request 4216246776 [ HASH N(AUTH_FAILED) ]
```

Phase 1 Pre-Shared Key Mismatch

A mismatched pre-shared key can be a tough to diagnose. An error stating the fact that this value is mismatched is not printed in the log, instead this messages is shown:

Log output from the initiator:

```
charon: 09[ENC] invalid HASH_V1 payload length, decryption failed?
charon: 09[ENC] could not decrypt payloads
charon: 09[IKE] message parsing failed
```

Log output from the responder:

```
charon: 09[ENC] invalid ID_V1 payload length, decryption failed?
charon: 09[ENC] could not decrypt payloads
charon: 09[IKE] message parsing failed
```

When the above log messages are present, check the Pre-Shared Key value on both sides to ensure they match.

Phase 1 Encryption Algorithm Mismatch

Log output from the initiator:

```
charon: 14[ENC] parsed INFORMATIONAL_V1 request 3851683074 [ N(NO_PROP) ]
charon: 14[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 14[CFG] received proposals: IKE:AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_
↪1024
charon: 14[CFG] configured proposals: IKE:AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_
↪1024
charon: 14[IKE] no proposal found
charon: 14[ENC] generating INFORMATIONAL_V1 request 3851683074 [ N(NO_PROP) ]
```

In this case, the log entry tells shows the problem exactly: The initiator was set for AES 128 encryption, and the responder is set for AES 256. Set both to matching values and then try again.

Phase 1 Hash Algorithm Mismatch

Log output from the initiator:

```
charon: 10[ENC] parsed INFORMATIONAL_V1 request 2774552374 [ N(NO_PROP) ]
charon: 10[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 14[CFG] received proposals: IKE:AES_CBC_256/MODP_1024
charon: 14[CFG] configured proposals: IKE:AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_
↪1024
charon: 14[IKE] no proposal found
charon: 14[ENC] generating INFORMATIONAL_V1 request 2774552374 [ N(NO_PROP) ]
```

The Hash Algorithm is indicated by the **HMAC** portion of the logged proposals. As can be seen above, the received and configured propsals do not have matching HMAC entries.

Phase 1 DH Group Mismatch

Log output from the initiator:

```
charon: 11[ENC] parsed INFORMATIONAL_V1 request 316473468 [ N(NO_PROP) ]
charon: 11[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 14[CFG] received proposals: IKE:AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_
↪8192
charon: 14[CFG] configured proposals: IKE:AES_CBC_256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_
↪1024
charon: 14[IKE] no proposal found
charon: 14[ENC] generating INFORMATIONAL_V1 request 316473468 [ N(NO_PROP) ]
```

DH group is indicated by the "MODP" portion of the listed proposal. As indicated by the log messages, the initiator was set for 8192 (Group 18) and the responder was set for 1024 (Group 2). This error can be corrected by setting the DH group setting on both ends of the tunnel to a matching value.

Phase 2 Network Mismatch

In the following example, the Phase 2 entry on the initiator side is set for 10.3.0.0/24 to 10.5.0.0/24. The responder is not set to match as it lists 10.5.1.0/24 instead.

Log output from the initiator:

```
charon: 08[CFG] proposing traffic selectors for us:
charon: 08[CFG] 10.3.0.0/24|/0
charon: 08[CFG] proposing traffic selectors for other:
charon: 08[CFG] 10.5.0.0/24|/0
charon: 08[ENC] generating QUICK_MODE request 316948142 [ HASH SA No ID ID ]
charon: 08[NET] sending packet: from 198.51.100.3[500] to 203.0.113.5[500] (236 bytes)
charon: 08[NET] received packet: from 203.0.113.5[500] to 198.51.100.3[500] (76 bytes)
charon: 08[ENC] parsed INFORMATIONAL_V1 request 460353720 [ HASH N(INVAL_ID) ]
charon: 08[IKE] received INVALID_ID_INFORMATION error notify
```

Log output from the responder:

```
charon: 08[ENC] parsed QUICK_MODE request 2732380262 [ HASH SA No ID ID ]
charon: 08[CFG] looking for a child config for 10.5.0.0/24|/0 === 10.3.0.0/24|/0
charon: 08[CFG] proposing traffic selectors for us:
charon: 08[CFG] 10.5.1.0/24|/0
charon: 08[CFG] proposing traffic selectors for other:
charon: 08[CFG] 10.3.0.0/24|/0
charon: 08[IKE] no matching CHILD_SA config found
charon: 08[IKE] queueing INFORMATIONAL task
charon: 08[IKE] activating new tasks
charon: 08[IKE] activating INFORMATIONAL task
charon: 08[ENC] generating INFORMATIONAL_V1 request 1136605099 [ HASH N(INVAL_ID) ]
```

In the responder logs it lists both the networks it received ("child config" line in the log) and what it has configured locally ("proposing traffic selectors for…" lines in the log). By comparing the two, a mismatch can be spotted. The "no matching CHILD_SA config found" line in the log will always be present when this mismatch occurs, and that directly indicates that it could not find a Phase 2 definition to match what it received from the initiator.

Phase 2 Encryption Algorithm Mismatch

Log output from the initiator:

```
charon: 14[CFG] configured proposals: ESP:AES_CBC_128/HMAC_SHA1_96/NO_EXT_SEQ
charon: 14[ENC] generating QUICK_MODE request 759760112 [ HASH SA No ID ID ]
charon: 14[NET] sending packet: from 198.51.100.3[500] to 203.0.113.5[500] (188 bytes)
charon: 14[NET] received packet: from 203.0.113.5[500] to 198.51.100.3[500] (76 bytes)
charon: 14[ENC] parsed INFORMATIONAL_V1 request 1275272345 [ HASH N(NO_PROP) ]
charon: 14[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 13[CFG] selecting proposal:
charon: 13[CFG] no acceptable ENCRYPTION_ALGORITHM found
```

(continues on next page)

(continued from previous page)

```
charon: 13[CFG] received proposals: ESP:AES_CBC_128/HMAC_SHA1_96/NO_EXT_SEQ
charon: 13[CFG] configured proposals: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
charon: 13[IKE] no matching proposal found, sending NO_PROPOSAL_CHOSEN
charon: 13[IKE] queueing INFORMATIONAL task
charon: 13[IKE] activating new tasks
charon: 13[IKE] activating INFORMATIONAL task
charon: 13[ENC] generating INFORMATIONAL_V1 request 1275272345 [ HASH N(NO_PROP) ]
```

In this case, the initiator receives a message that the responder could not find a suitable proposal ("received NO_PROPOSAL_CHOSEN"), and from the responder logs it is obvious this was due to the sites being set for different encryption types, AES 128 on one side and AES 256 on the other.

Phase 2 Hash Algorithm Mismatch

Log output from the initiator:

```
charon: 10[CFG] configured proposals: ESP:AES_CBC_256/HMAC_SHA2_512_256/NO_EXT_SEQ
charon: 10[ENC] generating QUICK_MODE request 2648029707 [ HASH SA No ID ID ]
charon: 10[NET] sending packet: from 198.51.100.3[500] to 203.0.113.5[500] (188 bytes)
charon: 10[NET] received packet: from 203.0.113.5[500] to 198.51.100.3[500] (76 bytes)
charon: 10[ENC] parsed INFORMATIONAL_V1 request 757918402 [ HASH N(NO_PROP) ]
charon: 10[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 11[CFG] selecting proposal:
charon: 11[CFG] no acceptable INTEGRITY_ALGORITHM found
charon: 11[CFG] received proposals: ESP:AES_CBC_256/HMAC_SHA2_512_256/NO_EXT_SEQ
charon: 11[CFG] configured proposals: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
charon: 11[IKE] no matching proposal found, sending NO_PROPOSAL_CHOSEN
charon: 11[IKE] queueing INFORMATIONAL task
charon: 11[IKE] activating new tasks
charon: 11[IKE] activating INFORMATIONAL task
charon: 11[ENC] generating INFORMATIONAL_V1 request 757918402 [ HASH N(NO_PROP) ]
```

Similar to a Phase 1 Hash Algorithm mismatch, the HMAC values in the log entries do not line up. However the responder also logs a clearer message "no acceptable INTEGRITY_ALGORITHM found" when this happens in Phase 2.

Phase 2 PFS Mismatch

Log output from the initiator:

```
charon: 06[ENC] generating QUICK_MODE request 909980434 [ HASH SA No KE ID ID ]
charon: 06[NET] sending packet: from 198.51.100.3[500] to 203.0.113.5[500] (444 bytes)
charon: 06[NET] received packet: from 203.0.113.5[500] to 198.51.100.3[500] (76 bytes)
charon: 06[ENC] parsed INFORMATIONAL_V1 request 3861985833 [ HASH N(NO_PROP) ]
charon: 06[IKE] received NO_PROPOSAL_CHOSEN error notify
```

Log output from the responder:

```
charon: 08[CFG] selecting proposal:
charon: 08[CFG] no acceptable DIFFIE_HELLMAN_GROUP found
```

(continues on next page)

(continued from previous page)

```
charon: 08[CFG] received proposals: ESP:AES_CBC_256/HMAC_SHA1_96/MODP_2048/NO_EXT_SEQ
charon: 08[CFG] configured proposals: ESP:AES_CBC_256/HMAC_SHA1_96/NO_EXT_SEQ
charon: 08[IKE] no matching proposal found, sending NO_PROPOSAL_CHOSEN
charon: 08[ENC] generating INFORMATIONAL_V1 request 3861985833 [ HASH N(NO_PROP) ]
```

Perfect Forward Secrecy (PFS) works like DH Groups on Phase 1, but is optional. When chosen PFS options do not match, a clear message is logged indicating this fact: "no acceptable DIFFIE_HELLMAN_GROUP found".

Note: In some cases, if one side has PFS set to *off* , and the other side has a value set, the tunnel may still establish and work. The mismatch shown above may only be seen if the values mismatch, for example *1* vs. *5*.

Mismatched Identifier with NAT

In this case, pfSense is configured for a **Peer Identifier** of *Peer IP address*, but the remote device is actually behind NAT. In this case strongSwan expects the actual private before-NAT IP address as the identifier. The racoon daemon used on older versions was much more relaxed and would match either address, but strongSwan is more formal and requires a correct match.

Log output from the responder:

```
   charon: 10[IKE] remote host is behind NAT
   charon: 10[IKE] IDir '192.0.2.10' does not match to '203.0.113.245'
[...]
   charon: 10[CFG] looking for pre-shared key peer configs matching 198.51.100.50...
↪203.0.113.245[192.0.2.10]
```

To correct this condition, change the **Peer Identifier** setting to *IP Address* and then enter the pre-NAT IP address, which in this example is `192.0.2.10`.

Disappearing Traffic

If IPsec traffic arrives but never appears on the IPsec interface (*enc0*), check for conflicting routes/interface IP addresses. For example, if an IPsec tunnel is configured with a remote network of `192.0.2.0/24` and there is a local OpenVPN server with a tunnel network of `192.0.2.0/24` then the ESP traffic may arrive, strongSwan may process the packets, but they never show up on *enc0* as arriving to the OS for delivery.

Resolve the duplicate interface/route and the traffic will begin to flow.

IPsec Status Page Issues

If the IPsec status page prints errors such as:

```
Warning: Illegal string offset 'type' in /etc/inc/xmlreader.inc on line 116
```

That is a sign that the incomplete xmlreader XML parser is active, which is triggered by the presence of the file */cf/conf/use_xmlreader*. This alternate parser can be faster for reading large *config.xml* files, but lacks certain features necessary for other areas to function well. Removing */cf/conf/use_xmlreader* will return the system to the default parser immediately, which will correct the display of the IPsec status page.

33.30 Troubleshooting Duplicate IPsec SA Entries

In certain cases an IPsec tunnel may show what appear to be duplicate IKE (Phase 1) or Child (Phase 2) security association (SA) entries.

After lengthy testing and research, the main way this starts to happen is when both sides negotiate or renegotiate simultaneously. If both peers initiate, reauthenticate, or rekey Phase 1 at the same time, it can result in duplicate IKE SAs. If both peers rekey Phase 2 at the same time, it can result in duplicate child SAs.

Mitigating this problem involves ensuring that the chance of simultaneous negotiation is minimized or eliminated. The easiest way to reach that goal is to set higher Phase 1 and Phase 2 lifetimes on one peer, or at least make sure both sides are **not** set identically.

Specific values vary but the settings below are the best general advice. Using the suggested values may not be possible due to peer constraints, such as third party vendors which do not support these settings or insist upon other settings. Variations are mentioned which should accommodate most situations, however, use as many strategies as possible.

Note: Most IPsec examples in the documentation do not yet reflect these values, as not all implementations are affected in the same way. However, over time they will be updated to match.

33.30.1 Current Version (2.5.0 and later)

The values in this section can be found in the current GUI. Older versions may have different values or some settings may not be available. Always run the most recent released version to ensure the best experience.

Peer A

Phase 1 (IKE SA)

Key Exchange Version *IKEv2* if supported by both peers.

Life Time The total IKE SA lifetime as a hard upper limit (e.g. `28800`)

Rekey Time 90% of total IKE SA lifetime (e.g. `25920`).

Reauth Time `0` to disable reauthentication.

If the peer requires IKEv1 or only supports IKEv2 reauthentication, set this as mentioned in **Rekey Time** above and also enable **Make Before Break** on the **Advanced Settings** tab.

Rand Time Defaults to 10% of IKE SA **Life Time** (e.g. `2880`). A larger **Rand Time** will decrease the chances of both peers renegotiating simultaneously.

Child SA Close Action *Restart/Reconnect* so that this side will reconnect child SA entries when they expire or fail.

Phase 2 (Child SA)

Life Time Total Child SA lifetime (e.g. `3600` for 1 hour). This peer will attempt to rekey the Child SA before it reaches this limit.

Rekey Time Leave blank to automatically use 90% of the **Life Time**, or choose a lower amount.

Rand Time Defaults to 10% of Child SA **Life Time** (e.g. `360`). A larger **Rand Time** will decrease the chances of both peers renegotiating simultaneously.

Peer B

Phase 1 (IKE SA)

Key Exchange Version *IKEv2* if supported by both peers.

Life Time The total IKE SA lifetime as a hard upper limit, but use a higher lifetime than Peer A by at least 10% (e.g. `31680`). With this peer set higher, Peer A will primarily manage IKE SA renegotiation, reducing the chance of conflicts.

Note: If the remote peer insists their lifetime be set to a specific value, then set peer A lower instead by a similar margin.

Rekey Time 90% of total IKE SA **Life Time**

Reauth Time Blank (disabled) to disable reauthentication.

If the peer requires IKEv1 or only supports IKEv2 reauthentication, set this as mentioned in **Rekey Time** above and also enable **Make Before Break** on the **Advanced Settings** tab.

Rand Time Defaults to 10% of IKE SA **Life Time** (e.g. `3168`). A larger **Rand Time** will decrease the chances of both peers renegotiating simultaneously. If using the same **Life Time** as Peer A, then increase this value further. If using a larger **Life Time** for Peer B, then leave this at the default or disable it (`0`).

Responder Only *Checked* so that this side **will not** automatically initiate IKE SA negotiation.

Note: This peer can still manually initiate a connection from **Status > IPsec**, but it won't happen automatically.

Child SA Close Action *Close connection and clear SA* so that when a Child SA expires, this side will remove the SA and not attempt to renegotiate a new entry.

Phase 2 (Child SA)

Life Time A larger value than the **Life Time** set on Peer A by at least 10%. For example, if Peer A is set to `3600`, set this to `5400`. That way Peer A will primarily manage Child SA renegotiation.

Note: If the remote peer insists their lifetime be set to a specific value, then set peer A lower instead by a similar margin.

Rekey Time Leave blank to automatically use 90% of the **Life Time**, or choose a lower amount.

Rand Time Defaults to 10% of Child SA **Life Time** (e.g. `540`). A larger **Rand Time** will decrease the chances of both peers renegotiating simultaneously. If using the same **Life Time** as Peer A, then increase this value further. If using a larger **Life Time** for Peer B, then leave this at the default or disable it (`0`).

Advanced IPsec Settings (both)

Make Before Break *Checked* if Phase 1 uses IKEv2 and reauthentication, not relevant otherwise.

33.30.2 Version 2.4.5-p1 and Before

The settings are almost all the same as the section above, with a couple changes. The primary difference is in the GUI settings for rekey and reauth. Only the differences are noted below, so follow the previous section except for the values noted here.

Warning: On 2.4.5-p1, setting **Responder Only** in the Phase 1 options requires an extra patch which can be applied using the *System Patches Package*: `9a69dd4b8ff6eeeaf5779b7388a10743afae8e91`.

Peer A

Lifetime The total time at which this peer will renegotiate the IKE SA (e.g. `28800`)

Margin Time An amount of time, in seconds, before the **Life Time** is reached when renegotiation begins. Defaults to `540`, but larger values can help reduce the chance of simultaneous renegotiation. Due to the default behavior of the IPsec daemon, this time can be randomly increased up to twice its value to further help avoid both sides choosing the same time. A larger value will help avoid potential collisions.

Disable Rekey *Unchecked*

Disable Reauth *Checked*

Peer B

Lifetime A higher time than the same field on Peer A by at least 10%. (e.g. `32400`)

Margin Time If using the same **Life Time** as Peer A, use a larger value to help avoid simultaneous renegotiation. If using a larger **Life Time** value, then leave this blank or set to the same value as Peer A.

Disable Rekey *Unchecked*

Disable Reauth *Checked*

33.30.3 Other Notes

The strongSwan daemon introduces randomness into the renegotiation process which can help mitigate the problem, but still leaves it up to chance if both peers are using the exact same lifetime values. That is why setting one peer higher, beyond the randomness threshold, is a better practice. The randomness also explains why the problem can take a while to manifest in certain environments as the duplicates do not happen until both peers happen to land on the same random rekey time.

33.31 Troubleshooting L2TP

This section covers troubleshooting steps for the most common problems users encounter with L2TP.

33.31.1 Cannot connect

Check that firewall rules have been added to the external interface where the L2TP traffic enters the firewall. Also make sure the client is connecting to the interface IP address chosen on the L2TP settings.

33.31.2 Connected to L2TP but cannot pass traffic

Ensure firewall rules have been added to the **L2TP VPN** interface as described in *Configure firewall rules for L2TP clients*.

Also ensure the remote subnet across the VPN is different from the local subnet. It is not possible to reach a 192.168.1.0/24 network across the VPN when the local subnet where the client resides is also 192.168.1.0/24, traffic destined for that subnet will never traverse the VPN because it is on the local network. This is why it is important to choose a relatively obscure LAN subnet when using a VPN.

33.31.3 Connection Fails with a Windows Client

If the IPsec layer appears to complete, but no L2TP traffic passes, it is likely a known incompatibility between Windows and the strongSwan daemon used on pfSense®. There is currently no known workaround except to move the Windows system out from behind NAT, or to use a different style VPN such as IKEv2.

33.31.4 L2TP Traffic Blocked Outbound

In some cases, such as when combined with IPsec, L2TP traffic may also require special handling via floating rules. This appears as blocked traffic in the *outbound* direction in the firewall logs, showing an L2TP server interface.

If this happens, add a floating rule as follows:

- Navigate to **Firewall > Rules**, **Floating** tab
- Click  **Add** to add a new rule to the top of the list
- Set **Action** to *Pass*
- Check **Quick**
- Select *L2TP VPN* for the **Interface**
- Set **Direction** to *Out*
- Set **Protocol** to *TCP*
- Set **Source/Destination** as needed, or set to *any*
- Advanced Features:
  - Set **TCP Flags** to *Any flags*
  - Set **State Type** to *Sloppy State*

33.32 Troubleshooting Access when Locked Out of the Firewall

Under certain circumstances an administrator can be locked out of the GUI. Don't be afraid if this happens; there are a number of ways to regain control. Some methods are a little tricky, but it is nearly always possible to recover access. The worst-case scenarios require physical access, as anyone with physical access can bypass security measures.

> **Warning:** Let the tactics in this section be a lesson: Physical security of a firewall is **critical**, especially in environments where the firewall is physically located in a common area accessible to people other than authorized administrators.

Before taking any of these steps, try the *Default Username and Password*.

33.32.1 Forgotten Password

The firewall administrator password can easily be reset using the firewall console if it has been lost. Access the physical console (*Connect to the Console*) and use option `3` to reset the password. This option can also reset the admin account if it is disabled or expired.

After resetting the password, login with the *Default Username and Password*.

33.32.2 Forgotten Password with a Locked Console

If the console is password protected, all is not lost. It takes two reboots to accomplish, but the password can be reset with physical access to the console:

- Connect to the console
- Reboot the firewall
- Choose the *Boot Single User* option (`2`) from the loader menu with the ASCII logo
- Press `Enter` when prompted to start `/bin/sh`
- Remount all partitions as rewritable:

  For systems installed using UFS:

```
# /sbin/mount -a -t ufs
```

  For systems installed using ZFS:

```
# /sbin/mount -u /
# /sbin/zfs mount -a
# /sbin/nextboot -D
```

- Run the built-in password reset command:

```
# /etc/rc.initial.password
```

- Follow the prompts to reset the password
- Run `/sbin/reboot` to reboot.

When the firewall reboots, login with the *Default Username and Password*.

33.32.3 HTTP vs HTTPS Confusion

Ensure the client is connecting with the proper protocol, either HTTP or HTTPS. If one doesn't work, try the other. If the GUI has not been configured correctly, the firewall may be running the GUI on an unexpected port and protocol combination, such as:

- `http://<hostname>:443`
- `https://<hostname>:80`

To reset this from the console, reset the LAN interface IP Address, enter the same IP address, and the script will prompt to reset the GUI back to HTTP.

33.32.4 Blocked Access with Firewall Rules

If a remote administrator loses access to the GUI due to a firewall rule change, then access can still be obtained from the LAN side. The LAN rules cannot prevent access to the GUI unless the anti-lockout rule is disabled. The anti-lockout rule ensures that hosts on the LAN are able to access the GUI at all times, no matter what the other rules on the LAN interface block.

Having to walk someone on-site through fixing the rule from the LAN is better than losing everything or having to make a trip to the firewall location!

33.32.5 Locked Out by Too Many Failed Login Attempts

Attempting to login to the GUI or SSH and failing many times will cause the connecting IP address to be added to the lockout table.

To regain access, login successfully from another IP address and then manually remove the entry as follows:

- Navigate to **Diagnostics > Tables**
- Select *sshguard*
- Click  by the entry or entries for workstations to allow again.

The lockout table may also be cleared by the console or ssh in the shell:

```
pfctl -T flush -t sshguard
```

33.32.6 Remotely Circumvent Firewall Lockout with Rules

There are a few ways to manipulate the firewall behavior at the shell to regain access to the firewall GUI. The following tactics are listed in order of how easy they are and how much impact they have on the running system.

Add a rule with EasyRule

The easiest way, assuming the administrator knows the IP address of a remote client PC that needs access, is to use the `easyrule` shell script to add a new firewall rule. In the following example, the `easyrule` script will allow access on the WAN interface, from `x.x.x.x` (the client IP address) to `y.y.y.y` (presumably the WAN IP address) on TCP port `443`:

```
# easyrule pass wan tcp x.x.x.x y.y.y.y 443
```

Once the `easyrule` script adds the rule, the client will be able to access the GUI from the specified source address.

Add an allow all WAN rule from the shell

Another tactic is to temporarily activate an "allow all" rule on the WAN to let a client in.

Warning: An "allow all" style rule is dangerous to have on an interface connected to a public or untrusted network, such as a WAN interface connected to the Internet. **Do not forget to remove the rule added by this script**

To add an "allow all" rule to the WAN interface, run the following command at a shell prompt:

```
# pfSsh.php playback enableallowallwan
```

Once the administrator regains access and fixes the original issue preventing them from reaching the GUI, remove the "allow all" rule from the WAN.

Disable the Firewall

An administrator can (very temporarily) disable firewall rules by using the physical console or SSH.

Warning: This completely disables `pf` which disables firewall rules and NAT. If the network run by this firewall relies on NAT to function, which most do, then running this command will disrupt connectivity from the LAN to the Internet.

To disable the firewall, connect to the physical console or ssh and use option `8` to start a shell, and then type:

```
# pfctl -d
```

That command will disable the firewall, including all NAT functions. Access to the GUI is now possible from anywhere, at least for a few minutes or until a process on the firewall causes the ruleset to be reloaded (which is almost every page save or **Apply Changes** action). Once the administrator has adjusted the rules and regained the necessary access, turn the firewall back on by typing:

```
# pfctl -e
```

Manual Ruleset Editing

The loaded ruleset is retained in `/tmp/rules.debug`. If the administrator is familiar with PF ruleset syntax, they can edit that file to fix the connectivity issue and reload those rules:

```
# pfctl -f /tmp/rules.debug
```

After getting back into the GUI with that temporary fix, the administrator must perform whatever work is required in the GUI to make the fix permanent. When the rules are saved in the GUI, the temporary edit to `/tmp/rules.debug` will be overwritten.

33.32.7 Remotely Circumvent Firewall Lockout with SSH Tunneling

If remote access to the GUI is blocked by the firewall, but SSH access is allowed, then there is a relatively easy way to get in: SSH Tunneling.

If the GUI is on port `443`, set the SSH client to forward local port `443` (or `4443`, or another port) to remote port `localhost:443`. If the firewall GUI is on another port, use that as the target instead. Then point the browser to `https://localhost`. Add the port to the end of the URL if it differs from the default 443, for example `https://localhost:4443`. If the GUI is using HTTP, change the protocol on the URL to `http://`.

Fill out the options as shown in Figure *Setting Up a Port 443 SSH Tunnel in PuTTY*, then click **Add**.

Once the client connects and authenticates, the GUI is accessible from the redirected local port.

33.32.8 Locked Out Due to Squid Configuration Error

If a firewall administrator accidentally configures Squid to use the same port as the GUI, it can cause a race condition for control of the port, depending on which service (re)starts at a particular time. If Squid manages to get control of the port that the GUI wants, then the GUI will not be accessible to fix the configuration.

The following procedure may help to regain control.

- Connect to the firewall console with SSH or physical access
- Start a shell, option `8` from the console.
- Terminate the squid process:

  ```
  # /usr/local/etc/rc.d/squid.sh stop
  ```

  If that doesn't work, try this command instead:

  ```
  # killall -9 squid
  ```

  or:

  ```
  # squid -k shutdown
  ```

Once the squid process is fully terminated, use console menu option `11` to restart the GUI process, and then attempt to access the GUI again.

Note: Work quickly or repeat the shutdown command, as squid may be automatically restarted by its internal monitoring scripts depending on the method used to stop the process.

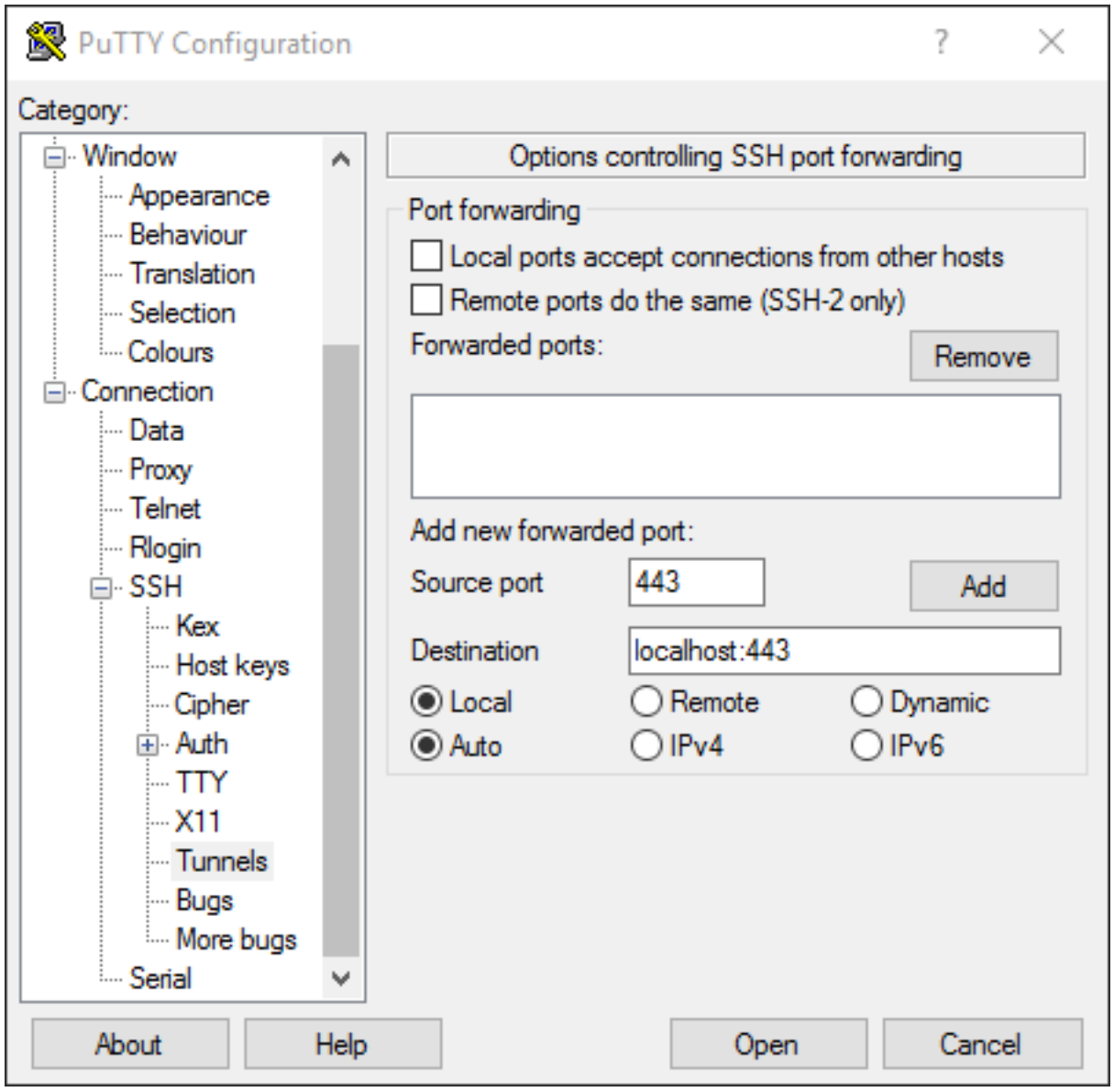


Fig. 4: Setting Up a Port 443 SSH Tunnel in PuTTY

33.32.9 Authentication Server Failure

LDAP and RADIUS authentication for the GUI automatically fall back to the local database if they fail. If the authentication server fails and all local accounts are disabled, locked out, passwords are not known, etc., then to get back in, regain access to the local `admin` account.

Connect to the console (*Connect to the Console*) or ssh and run option *3* to reset the credentials to the *Default Username and Password*.

33.33 Troubleshooting Blocked Log Entries for Legitimate Connection Packets

Sometimes log entries will be present that, while labeled with the "Default deny" rule, look like they belong to legitimate traffic. The most common example is seeing a connection blocked involving a web server.

Action	Time	Interface	Rule	Source	Destination	Protocol
✖	Sep 28 13:04:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:55632	45.57.40.1:443	TCP:FA
✖	Sep 28 13:04:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:55632	45.57.40.1:443	TCP:FA
✖	Sep 28 13:34:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:52185	45.57.41.1:443	TCP:FA
✖	Sep 28 13:34:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:52185	45.57.41.1:443	TCP:FA
✖	Sep 28 14:04:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:52194	45.57.41.1:443	TCP:FA
✖	Sep 28 14:04:44	LAN	Default deny rule IPv4 (1000000103)	172.[redacted]17:52194	45.57.41.1:443	TCP:FA

This is likely due to a TCP FIN packet arriving after firewall has removed the connection state. This happens because on occasion a packet will be lost, and the retransmits will be blocked because the firewall has already closed the connection. Another possible reason for the messages is if a packet arrived too slowly and was outside of its expected arrival window. It can also happen when web servers attempt to reuse connections.

In each case, the log entries are harmless and do not indicate a blocked connection. All stateful firewalls do this, though some do not generate log messages for this blocked traffic even if all blocked traffic is logged.

These blocked packets will occur even if rules exist which look as though they should match the traffic, such as an "Allow All" rule. Pass rules for TCP only allow **TCP SYN** packets to create a state. These rules assume TCP traffic with other flags will either be part of an existing state in the state table, or packets with spoofed TCP flags.

33.33.1 Asymmetric Routing

Blocked packets are also common for legitimate-looking traffic where routed networks and/or Multi-WAN are involved when *Asymmetric Routing* or other related causes are present in the network.

33.33.2 Clustering and Load Balancing

In a clustered environment, traffic arriving via the primary and leaving an internal interface can appear to be blocked on the secondary if the destination is a broadcast or multicast address like those used for Microsoft Network Load Balancing. The traffic appears to be blocked on the internal interface of the secondary from a public IP address source. Capturing the traffic on the secondary and inspecting the destination address in Wireshark will reveal the nature of the destination MAC address.

See also:

TCP Flags

33.34 Troubleshooting ARP Move Log Messages

pfSense® log entries may appear in the system log showing something similar to the following:

```
pfsense kernel: arp: 192.168.1.50 moved from c4:0c:5c:69:6c:05 to 62:1e:3e:43:04:0c␣
↪on em1
```

This indicates that the firewall saw the specified IP address move between the first MAC address and the second. This can happen for several reasons.

IP address conflict Two hosts are configured with the same IP address

ARP poisoning Someone on the network is ARP poisoning hosts

NIC teaming Some NIC teaming or bonding configurations will routinely log messages such as this because of the way they function. In these cases, this message is normal.

IP address moved to a different host or NIC If an actively used IP address is reassigned to a different device or different NIC, this message will be logged. This will only occur when an active IP is moved, for instance an expired DHCP lease that later is assigned to a different host will not trigger this as the IP must have an active ARP table entry on the firewall for this to occur.

Apple Bonjour sleep proxy Apple's Bonjour sleep proxy will cause these logs to appear because of its network behavior. If both of the listed MAC addresses are Apple vendor MACs, this is likely why and can be disregarded as normal behavior.

This logging can be disabled by setting the tunable `net.link.ether.inet.log_arp_movements` to value `0` under **System > Advanced**, **System Tunables** tab.

33.35 Troubleshooting "login on console as root" Log Messages

Occasionally, the following messages may appear in the system log:

```
login: login on console as root
```

or:

```
login: login on ttyv0 as root
```

This is normal. It means that the console menu stopped and restarted, or someone pressed enter (didn't choose a menu option) at the console menu. To suppress these messages, enable password protection for the console login and then it will only login after authentication. If console logins are already enabled, then this means someone logged into the console.

To password protect the console:

- Click **System > Advanced**
- Find the **Console Options** section near the end of the page
- Check **Password protect the console menu**
- Click **Save**

33.36 Troubleshooting "promiscuous mode enabled" Log Messages

The following log messages are recorded when a utility has placed the network card into "promiscuous mode":

```
Feb 10 01:41:58    kernel: vr0: promiscuous mode disabled
Feb 10 01:41:57    kernel: vr0: promiscuous mode enabled
Feb 10 01:41:54    kernel: vr0: promiscuous mode disabled
Feb 10 01:41:54    kernel: vr0: promiscuous mode enabled
Feb 10 01:41:50    kernel: vr0: promiscuous mode disabled
```

Promiscuous mode is a mode where the network card will receive every packet on the interface, regardless of the target MAC address, in order to monitor traffic. This is normal if for utilities such as *tcpdump*.

33.37 Troubleshooting Low Interface Throughput

In situations where the firewall is not transferring as much data as desired. There are many potential causes for this condition, most of which are listed here along with possible resolutions

33.37.1 Insufficient Hardware

The first thing to check is that the hardware is capable of pushing the expected amount of traffic. In some cases this is more obvious, such as a newer multi-core server being unable to transfer small amounts of packets, or an older firewall not being able to transfer high loads. Other cases are more subtle and require some testing and verification.

The most obvious test is to watch the firewall CPU load while transferring data. This can be observed from **Diagnostics > System Activity** or from the shell by running:

```
top -aSH
```

If an IRQ process for a network card is using a significant amount of CPU on a core, then either the hardware is being fully (or over) utilized, or the driver may need adjustments to work as expected. If the firewall is not under any stress whatsoever while transferring data, the problem likely lies elsewhere.

If the amount of "System" CPU is high and the amount of interrupts is low, the problem may be in the amount of packet processing happening in pf or being used for encryption. If pf is pushing the CPU as high as it can, it may require a faster CPU.

If the CPU is being used for encryption, a faster cipher may be chosen, or in some cases a cryptographic accelerator may be utilized.

33.37.2 Hardware/Driver Tuning Required

If a CPU core is fully utilized by interrupts, the network card driver may need tuning. Most of these tweaks are covered on *Tuning and Troubleshooting Network Cards*. Some cards, such as igb, are able to use more queues for processing packets which will spread the load across multiple cores and result in higher throughput, but not every workload is helped by these options, so less queues may also help.

Another item to check is under **System > Advanced** on the **Networking** tab. Ensure that the boxes are checked for **Disable hardware TCP segmentation offload** and **Disable hardware large receive offload**. If they are already checked, try toggling **Disable hardware checksum offload**. If no difference is observed, toggle it back.

33.37.3 Duplex Mismatch

A duplex mismatch is also possible, though this is more common on circuits 100Mbit/s or less. Some providers are stuck in the stone age and still insist on hard-coding ports on CPEs such as fiber converters at *100Mbit/s full-duplex*. If the CPE is hard-coded but the firewall is not, it would show as using 100Mbit/s half-duplex on **Status > Interfaces**. The duplex mismatch will lead to interface errors, collisions, and low throughput. Setting the speed and duplex is covered on *Forcing Interface Speed or Duplex Settings*.

33.37.4 Traffic Shaping

If the traffic shaping wizard was run previously before an increase in upstream bandwidth, the old limits may still be in effect. Visit **Firewall > Traffic Shaper** and check the root interface queues, and *qInternet* queues to ensure that any listed interface bandwidths are appropriately specified and current.

Also check the **Limiters** tab under the traffic shaper settings, verify that any configured limiters are set for appropriate speeds. Limiters may also need increased queue lengths to handle higher throughput volumes.

33.37.5 MTU Issues

Issues with upload speed frequently end up being issues with the MTU. If the MTU on pfSense® software (default 1500), is higher than the MTU of the upstream link, it can result in packets being fragmented, lost, or otherwise mishandled. Setting MSS clamping on the WANs or changing the MTU of the interface may help.

VPN + MTU Issues

Similar to the above, if large packets or high-throughput seems to break over a VPN, enable MSS Clamping for VPN Networks under **VPN > IPsec**, **Advanced Settings** tab. The default value for the option is `1400`, but try lower values such as `1350`, `1300`, `1250`, etc.

33.37.6 WAN Connection

There could also be issues between the WAN and the Modem/CPE. It could be a cable, or a quirk in how the two interfaces talk to each other. Place a small switch between the firewall and the Modem/CPE as a test.

33.37.7 Client/Testing Method

The slowness may not be from any cause on the firewall. It could be the client itself or how it connects. Testing a 100Mbit/s WAN over 802.11g wireless, for example, would never show full speed. Testing a 300Mbit/s WAN from a 100Mbit/s LAN connection would likewise not be a valid test.

Ensure the client is connected to the firewall through a connection at least as fast as the WAN supports.

33.37.8 ISP Issues

If every other factor has been eliminated, test the modem without the firewall involved. If the speed is still low, it may be the ISP to blame, or the Modem/CPE.

33.38 Troubleshooting Multi-WAN

This section describes some of the most common problems with multi-WAN and how to troubleshoot them.

33.38.1 Verify Firewall Rule Configuration

The most common error when configuring multi-WAN is improper firewall rules. Remember, the first matching rule wins and any further rules are ignored. If a policy routing rule is below the default LAN rule in the list, no traffic will ever match that rule because it will match the default LAN rule first. Review *Policy Routing Configuration* and verify the rules are correct.

If the rule ordering and configuration appears correct, it may help to enable logging on the rules. See *Troubleshooting Firewall Rules* for more information. Ensure the appropriate policy routing rule is passing the traffic.

33.38.2 Policy routing does not work for web traffic or all traffic

When a proxy package that can transparently capture HTTP traffic is used, such as squid, it overrides any policy routes that are defined for client traffic on that port. So no matter which gateway is set in firewall rules, traffic for HTTP (TCP port 80) will still go through squid and follow the firewall's default route.

33.38.3 Failover not working

If problems occur when an Internet connection fails, typically it is because the monitor IP address is still answering, so the firewall thinks the connection is still available. Check **Status > Gateways** to verify. An IP address on the modem may be used as a monitor IP address, which will still be accessible even if the Internet connection is down.

33.38.4 Load balancing not working

- Check that the Gateway Group is properly configured for load balancing, with at least two gateways on the same tier.
- Check that the firewall rules being matched direct traffic to the correct load balancing gateway group.
- Check that all of the gateways in the group show as Online under **Status > Gateways**. Connections marked as Offline will not be used.
- Check the testing methodology. Rather than testing with a web browser, try testing with curl or similar utilities which do not retain session data.
- Check that the traffic is not using a proxy or otherwise being initiated from a daemon on the firewall itself.

33.38.5 A gateway is incorrectly marked offline

If a gateway is listed as offline, but the WAN is actually up, several things could be at fault:

- First, test to see if the monitor IP address responds to a ping from a client device on the LAN, and again from **Diagnostics > Ping**.
- If the device with the monitor IP address or other intermediate hop drops ICMP echo request packets without a payload, manual pings would work but the gateway monitoring would fail. See *Advanced Gateway Settings* and set the payload to a value of `1` or higher.
- If the gateway or monitor IP address does not respond to ICMP echo requests, enter a different monitor IP address to use instead.
- If the monitor IP address is configured as a DNS server for a different WAN, the static routes could be causing a conflict and the echo requests to the gateway may not be following the expected path. Set a non-conflicting monitor IP address on the gateway.
- If there is an outbound NAT rule on the WAN with a **Source** of *any*, it can cause problems with traffic on the firewall, including monitoring traffic, because that will also NAT traffic from the firewall itself. This can be especially problematic if the source address is changed to a CARP VIP. Fix the outbound NAT.

If all else fails, it's possible the circuit really is down, but the testing methodology appears to show it up. Verify the Interface and Gateway settings and run the test again, and try `traceroute` to make sure the traffic is leaving using the expected path.

33.38.6 Ping works by IP address, but web browsing fails

In this case, the most likely cause is DNS. If the firewall DNS settings do not match those in *Interface and DNS Configuration*, clients may not be able to resolve DNS when a WAN is down. Review the settings and fix any problems that are found.

33.38.7 Squid doesn't seem to be using both connections

Squid and most other packages on the firewall itself do not understand load balancing; They will use only the WAN connection with the default gateway.

Check the Netgate Forum for package-specific alternate techniques.

33.39 Troubleshooting NAT

NAT can be a complex animal and in all but the most basic environments there are bound to be issues obtaining a good working configuration. This section will go over a few common problems and suggestions on how they can potentially be solved.

See also:

pfSense Hangouts on Youtube to view the May 2016 hangout for NAT on pfSense® software version 2.3, The June 2016 hangout on Connectivity Troubleshooting, and the December 2013 Hangout on Port Forward Troubleshooting, among others.

33.39.1 Port Forward Troubleshooting

Port Forwards in particular can be tricky, since there are many things to go wrong, many of which could be in the client configuration and not pfSense. Most issues encountered by users have been solved by one or more of the following suggestions.

Port forward entry incorrect

Before any other troubleshooting task, verify the settings for the port forward. Go over the process in *Adding Port Forwards* again, and double check that the values are correct. Remember, if the NAT IP address or the ports are changed, the firewall rule may also need adjusting if a linked firewall rule was not chosen.

Common things to check for:

- Correct interface: Usually WAN, or wherever traffic will enter the firewall.
- Correct NAT IP: The IP address must be reachable from an interface on the firewall.
- Correct port range: It must correspond to the service being forwarded.
- Source and source port should almost always be set to *any*.

Missing or incorrect firewall rule

After checking the port forward settings, double check that the firewall rule has the proper settings. An incorrect firewall rule would also be apparent by viewing the firewall logs (*Viewing the Firewall Log*). Remember, the destination for the firewall rule is the internal IP address of the target system and not the address of the interface containing the port forward. See *Rules for NAT* for more details.

Firewall is enabled on the target machine

Another thing to consider is that pfSense may be forwarding the port properly, but a firewall on the target machine may be blocking the traffic. If there is a firewall on the target system, check its logs and settings to confirm whether or not the traffic is being blocked at that point.

pfSense is not the target system gateway

In order for pfSense to properly forward a port for a local system, pfSense must be the default gateway for the target system. If pfSense is not the gateway, the target system will attempt to send replies to port forward traffic out whatever system is the gateway, and then one of two things will happen: It will be dropped at that point since there would be no matching connection state on that router or it would have NAT applied by that router and then be dropped by the system originating the request since the reply is from a different IP address than the one to which the request was initially sent.

Target system has no gateway or cannot use pfSense as its gateway

A subset of the larger problem of the target machine gateway is when the device has no gateway, or is incapable of having a gateway. In these cases, work around that problem by switching to Hybrid or Manual Outbound NAT and crafting a rule on the LAN or other internal interface facing the local device. This rule would translate traffic from any source going to the target system on the target port.

For example, if there is a file server that does not support a gateway located at 10.3.0.6, switch to Hybrid Outbound NAT and create a rule like Figure *Manual Outbound NAT Rule for LAN Device with Missing Gateway* to reach it from outside the network. The file server will see the LAN IP address of the firewall as the source of the traffic, and since that is "local" to the server, it will respond properly.

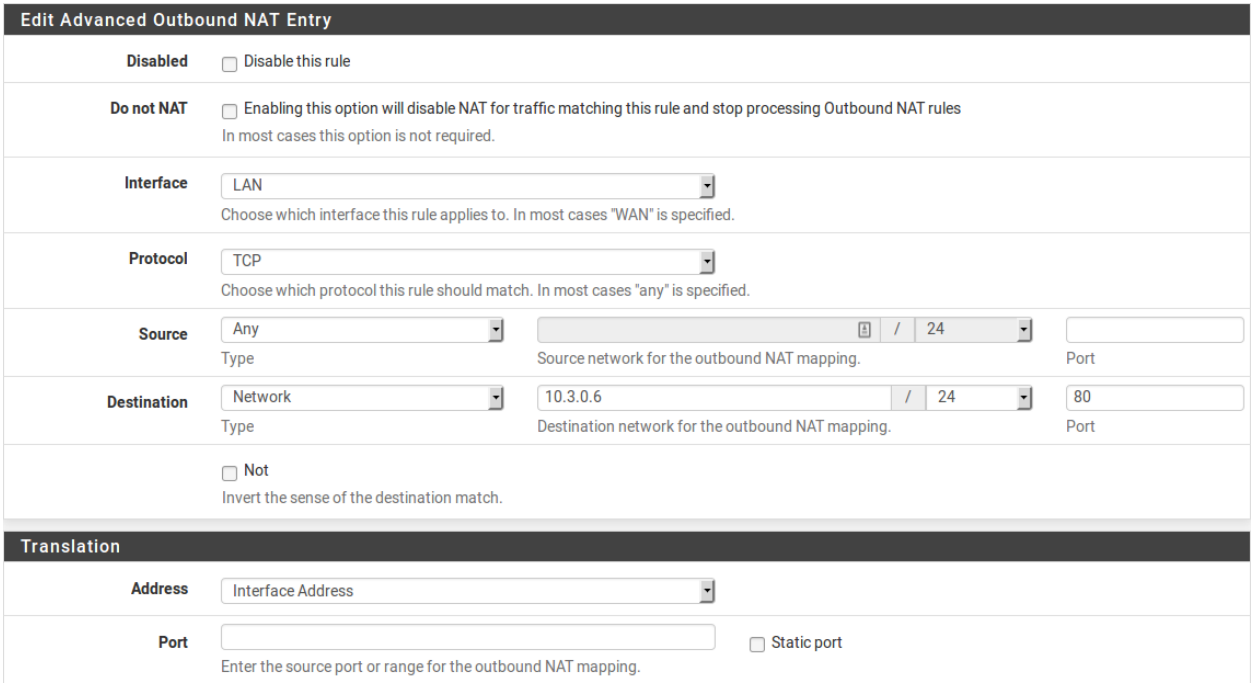


Fig. 5: Manual Outbound NAT Rule for LAN Device with Missing Gateway

Target machine is not listening on the forwarded port

If the request is rejected instead of timing out when the connection is tested, in all likelihood pfSense is forwarding the connection properly and the connection is rejected by the target system. This can happen when the target system has no service listening on the port in question, or if the port being forwarded does not match the port on which the target system is listening.

For example, if the target system is supposed to listen for SSH connections, but the port forward was entered for port 23 instead of 22, the request would most likely be rejected by the server. The difference can typically be detected by trying to connect to the port in question using netcat or telnet. A message such as "Connection refused" indicates something, frequently the inside host, is actively rejecting the connection. Using **Diagnostics > Test Port** can also help, see *Testing a TCP Port*.

ISP is blocking the port

Some ISPs filter incoming traffic to well-known ports. Check the Terms of Service (ToS) from the ISP to see if there is a clause about running servers. Such restrictions are more common on residential connections than commercial connections. When in doubt, a call to the ISP may clear up the matter.

If ports are being filtered by the ISP, moving the services to a different port may work around the restriction. For example, if the ISP disallows servers on port 80, try 8080 or 18080.

Before attempting to work around a filter, consult the ISP ToS to ensure that running a server is not a violation of their rules.

Testing from inside the network instead of outside

By default, port forwards will only work when connections are made from outside of the local network. This is a very common mistake when testing port forwards.

If port forwards are not required to work internally, see *NAT Reflection*. However, Split DNS (*Split DNS*) is a more proper and elegant solution to this problem without needing to rely on NAT reflection or port forwards, and it would be worth the time to implement that instead.

Even with NAT reflection, testing from inside the network isn't necessarily indicative of whether it will work from the Internet. ISP restrictions, restrictions on devices upstream from the firewall, amongst other possibilities are not possible to see when testing from within the network.

Incorrect or missing Virtual IP address

When using IP addresses that are not the actual IP addresses assigned to an interface, a Virtual IP address must be used (VIPs, see *Virtual IP Addresses*). If a port forward on an alternate IP address is not working, a different type of VIP may be required. For example, a Proxy ARP type may be necessary instead of an "Other" type VIP.

When testing, also make sure that the client is connecting to the proper VIP.

pfSense is not the border/edge router

In some scenarios pfSense is an internal router and there are other routers between it and the Internet also performing NAT. In such a case, a port forward must also be entered on the edge router forwarding the port to pfSense, which will then use another port forward to get it to the local system.

Forwarding ports to a system behind Captive Portal

Forwarding ports to a host behind a captive portal needs special consideration. See *Port Forwards Behind Portal Only Work When Target Logs In* for details.

Further testing needed

If none of these solutions result in a working port forward, consult *Firewall States* to look for NAT states indicating that the connection has made it through the firewall, or *Packet Capturing* for information on using packet captures to diagnose port forwarding issues.

33.39.2 NAT Reflection Troubleshooting

NAT Reflection (*NAT Reflection*) is complex, and as such may not work in some advanced scenarios. We recommend using Split DNS instead (see *Split DNS*) in most cases. However, NAT Reflection on current pfSense releases works reasonably well for nearly all scenarios, and any problems are usually a configuration mistake. Ensure that it was enabled the right way, and make sure a large range of ports is not being forwarded unnecessarily.

NAT Reflection rules are also duplicated for each interface present in the system, so if a lot of port forwards and interfaces are in use, the number of reflectors can easily surpass the limits of the system. If this happens, an entry is printed in the system logs. Check the system logs for any errors or information.

Web Access is Broken with NAT Reflection Enabled

If an improperly specified NAT Port Forward is present on the firewall, it can cause problems when NAT Reflection is enabled. The most common way this problem arises is with a local web server, and port 80 is forwarded there with an improperly specified **External Address**.

If NAT Reflection is enabled and the **External Address** is set to *any* , any connection made on the firewall comes up as the local web server. To fix this, edit the Port Forward for the offending port, and change **External Address** to *Interface Address* instead.

If an external address of *any* is required, then NAT Reflection will not work, and Split DNS must be used instead.

33.39.3 Outbound NAT Troubleshooting

When manual outbound NAT is enabled and there are multiple local subnets, an outbound NAT entry is required for each. This applies especially if traffic must exit with NAT after coming into the pfSense router via a VPN connection such as OpenVPN.

One indication of a missing outbound NAT rule would be seeing packets leave the WAN interface with a source address of a private network. See *Packet Capturing* for more details on obtaining and interpreting packet captures.

33.40 Troubleshooting 1:1 NAT

If 1:1 NAT (or even Outbound NAT) is properly configured, but the system still appears to access sites like https://www.netgate.com/ip and http://www.ipchicken.com from main WAN IP Address on the pfSense® firewall, then squid or another web proxy is likely being used.

Even though 1:1 NAT is in place, the web requests are still proxied through squid, and thus originate from the pfSense firewall itself.

The easy way around this is to allow that system to bypass the proxy like so:

1. Click **Services > Proxy Server**
2. Type the local IP Address of the system in the **Do NOT proxy these IPs** text box
3. Click **Save**

To proxy the web traffic and verify the 1:1 mapping is working properly, find some other service to verify with, such as:

- Login to a remote system and watch the firewall logs or tcpdump.
- Initiate some traffic from the system and verify the traffic is originating from the proper IP Address.
- Access an HTTPS site that does not flow through the proxy.

33.41 Troubleshooting NAT Port Forwards

If problems are encountered while attempting a port forward using pfSense® software, try the following.

1. If the *Port Forwards* guide was not followed exactly, delete anything that has been tried and start from scratch with those instructions.
2. Port forwards do not work internally unless *NAT reflection* has been enabled. Always test port forwards from outside the network, such as from a system in another location, or from a 3G/4G device.
3. Edit the firewall rule that passes traffic for the NAT entry and enable logging. Save and Apply Changes. Then try to access it again from the outside. Check the firewall logs (**Status > System Logs**, **Firewall** tab) to see if the traffic shows as being permitted or denied.
4. Check the states table under **Diagnostics > States**, filter on the source, destination, or port number to see if any entries are present. If entries are present that appear to match the NAT performed by the port forward, then the firewall is accepting and translating the traffic properly, so look at internal issues (e.g. client firewalls, etc, see below.)
5. Use a Packet Capture or **tcpdump** to see what is happening on the wire. This is the best means of finding the problem, but requires the most networking expertise. Navigate to **Diagnostics > Packet Capture** to capture traffic, or use **tcpdump** from the shell. Start with the WAN interface, and use a filter for the appropriate protocol and port. Attempt to access from outside the network and see if it shows up. If not, the ISP may be blocking the traffic, or if Virtual IPs are involved they may have an incorrect configuration. If the traffic is seen on the WAN interface, switch to the inside interface and perform a similar capture. If the traffic is not leaving the inside interface, there is a NAT or firewall rule configuration problem. If it is leaving the interface, and no traffic is coming back from the destination machine, the target system's default gateway may be missing or incorrect, it may not be listening on that port, or it may have a local firewall (Windows Firewall, iptables) blocking the traffic. For certain types of traffic return traffic may be seen indicating the host is not listening on that port. For TCP, this would be a TCP RST. For UDP, it may be an ICMP Unreachable message.

33.41.1 Common Problems

1. NAT and firewall rules not correctly added (see *Port Forwards*)

   Tip: Do **NOT** set a source port

2. Firewall enabled on client machine
3. Client machine is not using pfSense as its default gateway
4. Client machine not actually listening on the port being forwarded
5. ISP or something upstream of pfSense is blocking the port being forwarded
6. Trying to test from inside the local network, need to test from an outside machine
7. Incorrect or missing Virtual IP configuration for additional public IP addresses

8. The pfSense router is not the border router. If there is something else between pfSense and the ISP, the port forwards and associated rules must be replicated there.
9. Forwarding ports to a server behind a Captive Portal. An IP bypass must be added both to and from the server's IP in order for a port forward to work behind a Captive Portal.
10. If this is on a WAN that is not the default gateway, make sure there is a gateway chosen on this WAN interface, or the firewall rules for the port forward would not reply back via the correct gateway.
11. If this is on a WAN that is not the default gateway, ensure the traffic for the port forward is NOT passed in via Floating Rules or an Interface Group. Only rules present on the WAN's interface tab under Firewall Rules will have the reply-to keyword to ensure the traffic responds properly via the expected gateway.
12. If this is on a WAN that is not the default gateway, make sure the firewall rule(s) allowing the traffic in do not have the box checked to disable reply-to.
13. If this is on a WAN that is not the default gateway, make sure the master reply-to disable switch is not checked under **System > Advanced**, on the **Firewall/NAT** tab.
14. WAN rules should NOT have a gateway set, so make sure that the rules for the port forward do NOT have a gateway configured on the actual rule.
15. If the traffic appears to be forwarding in to an unexpected device, it may be happening due to UPnP. Check **Status > UPnP** to see if an internal service has configured a port forward unexpectedly. If so, disable UPnP on either that device or on the firewall.

33.42 Troubleshooting NAT Reflection

If an improperly specified NAT Port Forward exists it can cause problems when *NAT Reflection is enabled*.

The most common way this issue arises is when there is a local web server, and port 80 on the WAN is forwarded there. When NAT Reflection is enabled, any connection made to an external web site comes up as the internal web site instead.

To fix this, edit the NAT Port Forward for the offending port, and change **External Address** to ***Interface Address*** instead of *any*.

If an external address of "any" is absolutely required, then NAT Reflection will not be possible on this firewall and *Split DNS* must be used instead.

33.43 Troubleshooting OpenVPN

If problems are encountered when trying to use OpenVPN, consult this section for information on troubleshooting common issues.

33.43.1 Check OpenVPN Status

The first place to look is **Status > OpenVPN**. The connection status for each VPN is shown there. If a VPN is connected, waiting, reconnecting, etc, it would be indicated on that screen. For more information, see *Checking the Status of OpenVPN Clients and Servers*.

33.43.2 Check Firewall Log

If a VPN connection does not establish, or does establish but does not pass traffic, check the firewall logs under **Status > System Logs** on the **Firewall** tab. If traffic for the tunnel itself is being blocked, such as traffic to the WAN IP address on port 1194, then adjust the WAN firewall rules accordingly. If traffic is blocked on the OpenVPN interface, add rules to the **OpenVPN** tab to allow traffic there.

33.43.3 Some hosts work, but not all

If traffic between some hosts over the VPN functions properly, but some hosts do not, this is commonly one of four things.

Missing, incorrect or ignored default gateway If the device does not have a default gateway, or has one pointing to something other than pfSense®, it does not know how to properly get back to the remote network on the VPN. Some devices, even with a default gateway specified, do not use that gateway. This has been seen on various embedded devices, including IP cameras and some printers. There isn't anything that can be done about that other than getting the software on the device fixed. This can be verified by running a packet capture on the inside interface of the firewall connected to the network containing the device. Troubleshooting with tcpdump is covered in *Examples of using tcpdump on the command line*. If traffic is observed leaving the inside interface on the firewall, but no replies come back, the device is not properly routing its reply traffic or potentially blocking it via local firewall on the device.

Incorrect subnet mask If the subnet in use on one end is `10.0.0.0/24` and the other is `10.254.0.0/24`, and a host has an incorrect subnet mask of `255.0.0.0` or `/8`, it will never be able to communicate across the VPN because it thinks the remote VPN subnet is part of the local network and hence routing will not function properly.

Host firewall If there is a firewall on the target host, it may not be allowing the connections.

Firewall rules on pfSense Ensure the rules on both ends allow the desired network traffic.

33.43.4 Check the OpenVPN logs

Browse to **Status > System Logs** and click the **OpenVPN** tab to view the OpenVPN logs. Upon connecting, OpenVPN will log messages similar to the following example:

```
openvpn[32194]: UDPv4 link remote: 1.2.3.4:1194
openvpn[32194]: Peer Connection Initiated with 192.168.110.2:1194
openvpn[32194]: Initialization Sequence Completed
```

Note: The number following `openvpn` will differ, it is the process ID of the OpenVPN process making the connection.

If the *link remote* and *Peer Connection Initialized* messages are not shown when trying to connect, the cause is likely either incorrect client configuration, so the client is not attempting to connect to the correct server, or incorrect firewall rules blocking the client's connection.

33.43.5 Ensure no overlapping IPsec connections

Because of the way IPsec ties into the FreeBSD kernel, any enabled IPsec connection matching the local and remote subnets that exists when IPsec is enabled (even if it is not up) will cause that traffic to never be routed across the OpenVPN connection. Any IPsec connections specifying the same local and remote networks must be disabled. If an IPsec tunnel has been recently disabled or removed, check that its SPD entries have been removed by looking at **Status > IPsec** on the **SPD** tab. If they are present, remove them from that screen.

33.43.6 Check the system routing table

Browse to **Diagnostics > Routes** and review the routes known by the firewall. For site-to-site VPNs, routes will be present for the remote network(s) to the appropriate *tun* or *tap* interface. If the routes are missing or incorrect, the **Local Network**, **Remote Network**, or custom options are not configured correctly. If a shared key setup is in use and not PKI, ensure that "push" commands are not being used.

33.43.7 Test from different vantage points

If the connection appears to be up according to the logs, but it doesn't work from the LAN, try it from the firewall itself. These tests may be easily performed using the **Diagnostics > Ping** page on the firewall.

First test using the inside interface being used for OpenVPN internal traffic connections (typically LAN) as the ping source. If that doesn't work, try again using the *default* source address so that the firewall will source the ping from the OpenVPN interface itself.

If the *default* ping works but the internal network ping does not, check the firewall rules and routes on the far side.

33.43.8 Trace the traffic with packet captures

Using packet captures to determine where the traffic is or isn't flowing is one of the most helpful troubleshooting techniques. Start with the internal interface (commonly LAN) on the side where the traffic is being initiated, progress to the *tun* interface on that firewall, then the *tun* interface on the remote firewall, and finally the inside interface on the remote firewall. Determining where the traffic is seen and where it isn't can help greatly in narrowing down where the problem is located. Packet capturing is covered in detail in *Packet Capturing*.

33.43.9 Routes will not push to a client

When attempting to use the **Local Network** setting or a `push` statement to push routes to a client, and the client isn't receiving them properly, a couple things could be happening:

- Check that an SSL/TLS server setup is used with a **Tunnel Network** larger than a `/30`. The `server` mode in OpenVPN only takes effect when using a subnet large enough to contain multiple clients, such as a `/24`.
- If the client is running on Windows 10 or similar, try running the client as Administrator. Some versions of the OpenVPN client require Administrator mode to apply routes to the client PC routing table.
- When using a shared key setup, pushing routes will not work. Use the **Remote Network** boxes or `route` statements on each side (both client and server) to direct traffic to subnets on the other end of the tunnel.

33.43.10 Why can't I ping some OpenVPN adapter addresses?

In SSL/TLS server mode using a *net30* style **Topology**, OpenVPN will not respond to ping on certain virtual addresses used solely for routing endpoints. Do not rely on pinging the OpenVPN endpoint addresses as a means of determining if the tunnel is passing traffic properly. Instead, ping something in the remote subnet, such as the LAN IP address of the server.

Note: This is not relevant when using a *subnet* style **Topology**

According to the *`OpenVPN FAQ`_*, in the section titled *Why does OpenVPN's "ifconfig-pool" option use a /30 subnet (4 private IP addresses per client) when used in TUN mode?*:

> As 192.168.1.5 is only a virtual IP address inside the OpenVPN server, used as an endpoint for routes, OpenVPN doesn't bother to answer pings on this address, while the 192.168.1.1 is a real IP address in the servers O/S, so it will reply to pings.

This may seem a little counter-intuitive, since on the server the `ifconfig` output looks similar to:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
      inet6 fe80::202:b3ff:fe03:8028%tun0 prefixlen 64 scopeid 0xc
      inet 192.168.100.1 --> 192.168.100.2 netmask 0xffffffff
      Opened by PID 27841
```

While the client shows:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
      inet6 fe80::202:b3ff:fe24:978c%tun0 prefixlen 64 scopeid 0xa
      inet 192.168.100.6 --> 192.168.100.5 netmask 0xffffffff
      Opened by PID 1949
```

In this case, .5 or .1. likely will not respond to *ping*. The .5 address will not respond because it is a virtual address, and .1 because there is no route to reach it directly. The .5 and .6 addresses are part of a /30 that goes from .4 to .7, and trying to ping .1 would go out the default route instead.

There are many cases where the far side of an OpenVPN tunnel will respond to ping, but not the local. This is also counter-intuitive, but works especially in cases where a site-to-site link is present. If the server shows its tun addresses as `x.x.x.1 -> x.x.x.2` and the client shows the reverse - `x.x.x.2 -> x.x.x.1`, then the far will respond to ping from both ends.

33.43.11 Cannot route to clients on an SSL/TLS site-to-site tunnel

If an SSL/TLS site-to-site tunnel is used and all of the routes appear correct but traffic still cannot flow properly, check the tunnel network size. If this is a site-to-site setup between only two locations, the tunnel network should be a `/30` so that it does not require `iroute` statements to reach client networks. See the note at *IPv4/IPv6 Tunnel Network* for more information. When connecting multiple sites to a single server instance, check the setup against *OpenVPN Site-to-Site Configuration Example with SSL/TLS*, especially the client-specific overrides and `iroutes`.

33.43.12 Client Specific Override iroute entry seems to have no effect

When configuring a site-to-site PKI OpenVPN setup, an `iroute` statement must be configured using the **Remote Network** fields on the **Client Specific Overrides** entry set for the common name of the client certificate.

First, ensure that the **common name** matches the certificate and that the internal route is being learned/added as it expected. The log verbosity in OpenVPN may need increased (i.e. *verb 10* in the custom options) to see if this is working.

Also, for each network used in a **Client Specific Override Remote Network** entry (`iroute`), a **Remote Network** (`route`) is required in the server as well. The **Remote Network** (`route`) definitions on the server settings are for the firewall operating system to know that the networks will be routed to OpenVPN from everywhere else. The **Remote Network** (`iroute`) options on the **Client Specific Override** entry are internal to OpenVPN so it knows which networks are routed to a specific certificate.

33.43.13 Why do OpenVPN clients all get the same IP address?

If the same certificate is used for all clients, which we strongly discourage, then the clients are all assigned the same IP address when they connect. To work around this, check **Duplicate Connections** on the server configuration.

33.43.14 Importing OpenVPN DH Parameters

When importing an existing OpenVPN setup into pfSense, there is no need to import DH Parameters. DH parameters are not specific to a given setup in the way that certificates or keys are.

To put it simply, the DH parameters are some extra bits of randomness that help out during the key exchange process. They do not have to match on both sides of the tunnel, and new ones can be made at any time. There is no need to import an existing set of DH parameters.

Note: By default, pfSense uses a set of pre-generated DH parameters. A new set may be generated manually if desired, see *DH Parameters Length* for details.

33.44 Troubleshooting OpenVPN Remote Access Client IP Address Assignments

If the same certificate has been used for multiple clients (which we do not recommend!), then all clients may be assigned the same IP address when they connect.

To work around this, duplicate connections must be allowed using one of the following methods:

- Check **Duplicate Connections** on the OpenVPN server configuration
- Add the following line to the custom options on the server:

```
duplicate-cn
```

33.45 Troubleshooting Windows OpenVPN Client Connectivity

Historically, OpenVPN clients on Windows had issues with routing due to lack of privileges. Current versions of the OpenVPN client for Windows run as a service which does not require administrative privileges. See *Installing the OpenVPN Client on Windows* for details. Older legacy clients may still need such privileges, but in most cases the correct answer is to upgrade OpenVPN.

When installing OpenVPN using the *OpenVPN Client Export Package* package it will not upgrade an existing installation of OpenVPN. To ensure the client is updated properly, uninstall the old client first.

33.46 Troubleshooting Windows/SMB Share Access from OpenVPN Clients

If Windows/SMB shares are not able to accessed by OpenVPN clients but other services work as desired, visit **VPN > OpenVPN**, edit the server in question, and *check* **Enable NetBIOS over TCP/IP**.

It may also be necessary to set *MSS clamping for VPNs*.

33.47 Troubleshooting OpenVPN Internal Routing (iroute)

When configuring a site-to-site PKI (SSL) OpenVPN setup, an internal route must be configured for the client subnet on the **Client Specific Overrides** tab set for the client certificate's **common name**, using either the **IPv4/IPv6 Remote Network/s** boxes or manually using an `iroute` statement in the advanced settings. Each network listed in the **IPv4 Remote Network/s** and **IPv4 Remote Network/s** boxes will have an `iroute` created automatically.

Next, ensure that the **common name** matches and that the internal route is being learned/added as it should be. Log verbosity in OpenVPN may need increased to see if this is working. On **Status > OpenVPN** the internal routing for the OpenVPN server may also be viewed while the client is connected.

For each network that needs an `iroute` statement, the **server** definition must also have the same network(s) listed as **IPv4/IPv6 Remote Networks** or as `route` statements in the advanced options box.

For example:

- Server1 custom options:

```
push "route a.a.a.0 255.255.255.0";
route b.b.b.0 255.255.255.0;
```

- Client Specific Overrides for **Common Name** `client1`:
  - **IPv4 Remote Network/s** set to `b.b.b.0/24` -OR-
  - Advanced options:

```
iroute b.b.b.0 255.255.255.0;
```

client1 custom options:

```
(blank -- no route statements needed)
```

On the server side, every `iroute` needs a corresponding `route`. The `route` entries are for the OS to know that the subnet(s) should be routed to OpenVPN from at the OS level. The `iroute` statements are internal to OpenVPN, so it knows which network goes to which client based on its certificate.

See Also:

- http://www.secure-computing.net/wiki/index.php/OpenVPN/Routing

33.48 Troubleshooting OpenVPN Push Routes

If it appears that OpenVPN will not push routes to a client, ensure that a *Multi-site style PKI/SSL setup* is in use and not a *shared key setup* or an SSL/TLS setup using a /30 tunnel network.

Routes cannot be pushed on a shared key setup or an SSL/TLS setup using a /30 tunnel network. Routes may be directly added to the client configuration using the **IPv4 Remote Networks** list (e.g. `192.168.99.0/24, 192.168.203.0/24`)

33.49 Troubleshooting Lost Traffic or Disappearing Packets

If there are issues with traffic being lost, or packets that seem to disappear or never show up (or leave) an interface, try disabling Checksum Offloading as follows:

- Navigate to **System > Advanced** on the **Networking** tab
- Check **Disable hardware checksum offload** under the **Network Interfaces** header.
- Click **Save**.

Then try to reproduce the problem.

Note: A reboot may be desired after applying this option, but it should not be necessary.

This has historically been an issue with Realtek NICs and in some scenarios with Via Rhine NICs. Also, some specific Intel fxp chips, as well as virtualized/emulated NICs in some hypervisors, especially those that use VirtIO.

The problem may also manifest as a PPPoE connection that will establish a login and bring up the interface, but will not pass traffic.

33.50 Troubleshooting a Broken pkg Database

In rare edge cases it is possible for the pkg database in **/var/db/pkg/** to become corrupted. In the unlikely event this happens to a firewall, it can usually be corrected by running a few commands to re-create the database.

Note: The following commands only account for the base system of a typical CE installation.

1. Ensure that the package database directory exists:

```
/bin/mkdir -p /var/db/pkg/ /root/var/db/pkg/
```

2. Force an update of the package repository data:

```
/usr/local/sbin/pkg-static update -f
```

3. Force a reinstall of the pfSense® base package and kernel:

```
/usr/local/sbin/pkg-static install -yf pkg pfSense pfSense-kernel-pfSense
```

4. Refresh the **php.ini** and other files to ensure they are loading the correct modules:

```
/etc/rc.php_ini_setup
```

5. If any additional packages were installed, reinstall them manually using the GUI if possible, or by using `pkg-static install` as above:

```
/usr/local/sbin/pkg-static install -yf <additional-package> <another-additional-
↪package>
```

33.51 Troubleshooting Routes

When diagnosing traffic flow issues, one of the first things to check is the routes known to pfSense®.

33.51.1 Viewing Routes

There are two ways to view the routes: Via the WebGUI, and via the command line.

To view the routes in the WebGUI, navigate to **Diagnostics > Routes** and output is shown similar to Figure *Route Display*.

IPv4 Routes

Destination	Gateway	Flags	Use	Mtu	Netif	Expire
default	198.51.100.1	UGS	1797	1500	igb1	
10.2.0.0/24	link#2	U	0	1500	igb0	
10.2.0.1	link#2	UHS	0	16384	lo0	
127.0.0.1	link#11	UH	204	16384	lo0	
198.51.100.0/24	link#3	U	907	1500	igb1	
198.51.100.1	00:08:a2:09:95:b6	UHS	2519	1500	igb1	
198.51.100.2	link#3	UHS	0	16384	lo0	

Fig. 6: Route Display

The output from the command line is similar to that seen in the WebGUI:

```
# netstat -rWn
Routing tables

Internet:
Destination        Gateway            Flags        Use     Mtu      Netif Expire
default            198.51.100.1       UGS         1822    1500       igb1
10.2.0.0/24        link#2             U              0    1500       igb0
10.2.0.1           link#2             UHS            0   16384        lo0
127.0.0.1          link#11            UH           204   16384        lo0
198.51.100.0/24    link#3             U           1181    1500       igb1
198.51.100.1       00:08:a2:09:95:b6  UHS         2789    1500       igb1
198.51.100.2       link#3             UHS            0   16384        lo0
```

The columns shown on these screens indicate various properties of the routes, and are explained later in this section.

Destination

This column contains the destination host or network. The default route for the system is simply listed as `default`. Otherwise, hosts are listed as by IP address, and networks are listed with an IP address and CIDR subnet mask.

Gateway

A gateway is the router through which packets going to a specific destination are sent. If this column shows a link, such as `link#1`, then that network is directly reachable by that interface and no special routing is necessary. If a host is visible with a MAC address, then it is a locally reachable host with an entry in the ARP table, and packets are sent there directly.

Flags

There are quite a few flags, all of which are covered in the FreeBSD man page for *netstat(1)*, reproduced in Table *Route Table Flags and Meanings* with some modifications.

Table 1: Route Table Flags and Meanings

Letter	Flag	Meaning
1	RTF_PROTO1	Protocol specific routing flag #1
2	RTF_PROTO2	Protocol specific routing flag #2
3	RTF_PROTO3	Protocol specific routing flag #3
B	RTF_BLACKHOLE	Discard packets during updates
b	RTF_BROADCAST	Represents a broadcast address
D	RTF_DYNAMIC	Created dynamically by redirect
G	RTF_GATEWAY	Destination requires forwarding by intermediary
H	RTF_HOST	Host entry (net otherwise)
L	RTF_LLINFO	Valid protocol to link address translation
M	RTF_MODIFIED	Modified dynamically (by redirect)
R	RTF_REJECT	Host or net unreachable
S	RTF_STATIC	Manually added
U	RTF_UP	Route usable
X	RTF_XRESOLVE	External daemon translates proto to link address

For example, a route flagged as `UGS` is a usable route, packets are sent via the gateway listed, and it is a static route.

Refs

This column counts the current number of active uses of a given route.

Use

This counter is the total number of packets sent via this route. This is helpful for determining if a route is actually being used, as it will continually increment as packets utilize the route.

Netif

The network interface used for this route.

Expire

For dynamic entries, this field shows how long until this route expires if it is not used again.

33.51.2 Using traceroute

Traceroute is a useful tool for testing and verifying routes and multi-WAN functionality, among other uses. It shows each "hop" along the path a packet travels from one end to the other, along with the latency encountered in reaching that intermediate point. On pfSense software, a traceroute can be performed by navigating to **Diagnostics > Traceroute**, or by using `traceroute` at the command line. From clients running Windows, the program is available under the name `tracert`.

Every IP packet contains a time-to-live (TTL) value. When a router passes a packet, it decrements the TTL by one. When a router receives a packet with a TTL of 1 and the destination is not a locally attached network, the router returns an ICMP error message "Time-to-live exceeded" and drops the packet. This is to limit the impact of routing loops, which otherwise would cause each packet to loop indefinitely.

Traceroute uses this TTL to its advantage to map the path to a specific network destination. It starts by sending the first packet with a TTL of 1. The first router (usually the default gateway) will send back an ICMP time-to-live exceeded error. The time between sending the packet and receiving the ICMP error is the time displayed, listed along with the IP address that sent the error and its reverse DNS, if any. After sending three packets with a TTL of 1 and displaying their response times, it will increment the TTL to 2 and send three more packets, noting the same information for the second hop. Traceroute increments the TTL and repeats the process until it reaches the specified destination, or exceeds the maximum number of hops.

Traceroute functions slightly differently on Windows and Unix-like operating systems (BSD, Linux, Mac OS X, Unix, etc.). Windows uses ICMP echo request packets (pings) while Unix-like systems use UDP packets by default. ICMP and UDP are layer 4 protocols, and traceroute is done at layer 3, so the protocol used is largely irrelevant except when considering a policy routing configuration. Traceroute from Windows clients will be policy routed based on which rule permits ICMP echo requests, while Unix-like clients will be routed by the rule matching the UDP ports in use.

In this example, `traceroute` is used to view the route to `www.google.com`:

```
# traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 74.125.95.99
traceroute to www.l.google.com (74.125.95.99), 64 hops max, 40 byte packets
 1  core (172.17.23.1)  1.450 ms  1.901 ms  2.213 ms
 2  172.17.25.21 (172.17.25.21)  4.852 ms  3.698 ms  3.120 ms
 3  bb1-g4-0-2.ipltin.ameritech.net (151.164.42.156)  3.275 ms  3.210 ms  3.215 ms
 4  151.164.93.49 (151.164.93.49)  8.791 ms  8.593 ms  8.891 ms
 5  74.125.48.117 (74.125.48.117)  8.460 ms  39.941 ms  8.551 ms
 6  209.85.254.120 (209.85.254.120)  10.376 ms  8.904 ms  8.765 ms
 7  209.85.241.22 (209.85.241.22)  19.479 ms  20.058 ms  19.550 ms
 8  209.85.241.29 (209.85.241.29)  20.547 ms  19.761 ms
    209.85.241.27 (209.85.241.27)  20.131 ms
 9  209.85.240.49 (209.85.240.49)  30.184 ms
    72.14.239.189 (72.14.239.189)  21.337 ms  21.756 ms
10  iw-in-f99.google.com (74.125.95.99)  19.793 ms  19.665 ms  20.603 ms
```

The output shows that it took 10 hops to get there, and the latency generally increased with each hop, which is expected.

Note: When utilizing policy routing, such as with Multi-WAN, the firewall itself may not appear as a hop in traceroute. When policy routing is employed, pf does not decrement the TTL when forwarding packets, so traceroute cannot detect it as an intermediate router.

33.51.3 Routes and VPNs

Depending on the VPN being used, a route may not display in the table for the far side. IPsec does not use the routing table, it is instead handled internally in the kernel using IPsec security policy database (SPD) entries. Static routes will never cause traffic to be directed across an IPsec connection. OpenVPN uses the system routing table and as such entries are present for networks reachable via an OpenVPN tunnel, as in the following example:

```
#netstat -rWn
Routing tables

Internet:
Destination        Gateway            Flags       Use    Mtu      Netif Expire
default            198.51.100.1       UGS       92421   1500        em0
10.6.0.0/16        10.6.203.1         UGS           0   1500      ovpnc2
10.6.203.0/24      10.6.203.2         UGS           0   1500      ovpnc2
10.6.203.1         link#9             UH            0   1500      ovpnc2
10.6.203.2         link#9             UHS           0  16384        lo0
10.7.0.0/24        link#2             U       1260771   1500        em1
10.7.0.1           link#2             UHS           0  16384        lo0
127.0.0.1          link#7             UH          866  16384        lo0
198.51.100.0/24    link#1             U       1251477   1500        em0
198.51.100.7       link#1             UHS           0  16384        lo0
```

The OpenVPN interface is `10.6.203.2`, with a gateway of `10.6.203.1` and the interface is `ovpnc2`. The network reachable using OpenVPN in this example is `10.6.0.0/16`.

With IPsec, traceroute is not as useful as with routed setups like OpenVPN, because the IPsec tunnel itself does not have IP addresses. When running traceroute to a destination across IPsec, a timeout will be shown for the hop that is the IPsec tunnel.

33.52 Troubleshooting in Single User Mode

Warning: This option must only be used under the guidance of a support representative or a FreeBSD user with advanced knowledge.

Single user mode is a special boot mode in which only the operating system kernel is loaded and a bare minimum of functions are available. For example, networking is not configured, the GUI is not available, and no system services are started. The root filesystem defaults to read-only and other filesystems are not mounted. In single user mode, commands must be entered into a shell prompt on the primary system console.

Warning: This action **requires** access to the primary system console, which depending on hardware and configuration may be a video console or serial console. Ensure console access is available and working before attempting to enter single user mode.

33.52.1 Entering Single User Mode

There are multiple ways to enter single user mode:

- From the console or SSH menu, choose option `5` *Reboot System* and then enter `S` (capital `s`) to reboot and automatically enter single user mode.

  Warning: ZFS systems will not automatically clear this configuration when exiting single user mode. See *ZFS Systems*.

- From the loader menu with the ASCII logo, choose the menu option to enter single user mode, typically option `2`
- Enter the loader directly, either by using option `3` from the ASCII loader menu or by pressing space while the kernel loads.

  From the `OK` prompt in the loader, enter `boot -s`

After invoking one of those options, the kenel will load and then immediately after, the operating system will prompt for a shell:

```
Enter full pathname of the shell or RETURN for /bin/sh:
```

At that point, press `Enter` and the resulting shell prompt is now running in single user mode.

Tip: Redminder: At this point, all filesystems are still read-only.

33.52.2 Exiting Single User Mode

There are two ways to exit single user mode, and the method to use depends on the changes made.

- Run `/sbin/reboot` or an equivalent command to force an operating system reboot. This is the safest choice as it will ensure the system is fully reinitialized. This method is required when making boot-time changes such as those in `/boot/loader.conf.local`.
- Exiting the single user mode shell via `exit`, `logout`, or `^D` (Ctrl-D) will terminate the single user mode shell and then continue to boot the system into its regular multi-user mode. This may be OK for simple changes such as `config.xml` alterations, but will not activate changes which must be present before the kernel loads.

33.52.3 Working in Single User Mode

Mounting Filesystems

In single user mode the root filesystem defaults to read-only and other filesystems are not mounted.

UFS Systems

To mount all UFS type filesystems in read-write mode, run the following command:

```
# /sbin/mount -a -t ufs
```

ZFS Systems

On ZFS systems the procedure to mount the all filesystems as read-write in single user mode requires several commands:

First, remount the root slice as read-write:

```
# /sbin/mount -u /
```

Next, mount all ZFS filesystems, datasets, etc:

```
# /sbin/zfs mount -a
```

Finally, if the system was put into single user mode using the reboot menu `S` option, the `nextboot` configuration **must** be cleared manually:

```
# /sbin/nextboot -D
```

Warning: Failing to clear the `nextboot` configuration will result in the system booting back into single user mode on every boot.

Tip: It is safe to run this command even if the system was not booted in that way, so always run this command on ZFS systems before exiting single user mode.

Running Commands

Be aware that when running commands single user mode, their behavior may be unexpected as the system does not have a full traditional terminal setup, and some default shell environment variables and console/terminal settings are not available. For example, using `vi`, the arrow keys on the keyboard may not work or the dimensions of the terminal may be incorrect.

Some commands may be run without typing the full pathname to the binary, but it is safest to use the full path where possible.

33.53 Troubleshooting Snort Rule Updates

33.53.1 MD5 Signature Mismatch

Periodically, Sourcefire redesigns their site or updates the engine and rules, and the snort package needs an update to accommodate this change. Removing and then installing the snort package again is required to restore proper functionality, assuming the package has been updated to match the upstream rule format.

33.53.2 Upstream Issues

Rule problems can almost always be solved by waiting 20-30 minutes and then trying the download again. Failing that, uninstall the package completely and then reinstall the package to ensure the snort binaries are the latest/correct ones.

33.53.3 Space Issues

If the */tmp* slice is small, because the firewall is running with */tmp* on a RAM disk, current rulesets can easily fill the slice up and cause numerous rule-related errors.

If there is sufficient RAM, increase the size of */tmp* using the options on **System > Advanced**, **Miscellaneous** tab.

33.54 Troubleshooting the Squid Package

33.54.1 Disk Usage Issues

The *swap.state* from Squid file can grow large and consume all available drive space. See *Tuning the Squid Package* for more details.

33.54.2 Sites not loading with splice / Error 409 in access log

As a security measure, squid will not allow a user to connect to a site that has a hostname that does not match its IP address. This prevents clients from hardcoding or altering DNS responses to evade access controls. The side effect of this, however, is that sites which employ round-robin DNS or other DNS optimizations can cause squid to block or drop connections those sites unintentionally. The squid access log will have a **409 (Conflict)** error code when a connection is dropped for this reason.

This happens with sites such as Google or Facebook when the client and squid use different sources for DNS, and thus get different DNS results for the same query because the results are randomized. Even though the address for the server is valid, the disparity causes squid to drop the connection.

The solution is to have the clients use the firewall as their DNS server, so that both squid and clients use the same DNS source and the results will match.

33.54.3 Clear Cache

Resetting the cache in squid can often clear up issues without performing a more complicated procedure. Before performing a full reset, try clearing and resetting the cache:

```
mv /var/squid/cache /var/squid/cache.old
squid -z
rm -rf /var/squid/cache.old
```

The old cache should be moved, then reset, and then the old cache should be removed, as above, because removing the cache directory can be time consuming, and if it is moved first, then its removal will not prevent squid from being run while it is happening.

33.54.4 Complete Reset

When troubleshooting squid/squidGuard there are some procedures that may be followed to ensure things are completely reset.

1. Remove the packages from **System > Packages** on the **Installed Packages** tab in the proper order: Lightsquid, SquidGuard, then Squid
2. Remove the contents of the squid directory and cache:

   ```
   rm -rf /var/squid
   ```

3. Remove the Squid and related package include files:

   ```
   rm /usr/local/pkg/*squid*
   rm -rf /usr/local/etc/squid/
   ```

4. Ensure any leftover PBI symlinks are removed:

   ```
   find / -type l -lname '/usr/pbi/*' -delete
   ```

5. [Optional] Remove the settings from inside **config.xml** using one of the following methods:
   - From **Diagnostics > Command Prompt** in the **PHP Execute** box:

     ```
     $squid_sections = array("squid", "squidnac", "squidcache", "squidauth",
     ↪"squidextauth", "squidtraffic", "squidupstream", "squidusers");
     foreach ($squid_sections as $sec) {
      if (is_array($config['installedpackages'][$sec]))
              unset($config['installedpackages'][$sec]);
     }
     write_config("Removed Squid");
     ```

   - Or to remove squid, squidguard, lightsquid, and anything else with 'squid' in its package name from **Diagnostics > Command Prompt** in the **PHP Execute** box:

     ```
     foreach (array_keys($config['installedpackages']) as $sec) {
      if (strpos($sec, "squid") !== false)
              unset($config['installedpackages'][$sec]);
     }
     write_config("Removed all squid-related settings");
     ```

   - Or backup **config.xml**, edit the settings out, then restore.
6. Navigate to **System > Packages** and on the **Available Packages** tab, reinstall the following packages in order:

1. Squid
2. squidGuard
3. Lightsquid

See also:

For assistance in solving problems, post on the Cache/Proxy category of Netgate Forum.

33.55 Troubleshooting Hardware Shutdown and Power Off

If a firewall device does not automatically power itself off, this is typically a case of FreeBSD and ACPI not working well together on a particular hardware combination.

As a test, enter this at the CLI then attempt a power-down:

```
sysctl hw.acpi.disable_on_reboot=1
```

For a more permanent solution, add an entry under **System > Advanced** on the **Tunables** tab to set:

```
hw.acpi.disable_on_reboot=1
```

33.56 Troubleshooting Clock Issues

Time and clock issues are relatively common on hardware, but on firewalls they are critical, especially if the firewall is performing tasks involving validating certificates as part of a PKI infrastructure.

Proper time synchronization is an absolute necessity on systems which do not have a battery onboard to preserve their date and time settings when power is removed.

Not only will getting this all in line help with critical system tasks, but it also ensures that the log files on the firewall are properly timestamped, which aids with troubleshooting, record keeping, and general system management.

33.56.1 Time Keeping Problems

Hardware can have significant problems keeping time, though such problems are generally isolated to older, low-quality hardware. All PC clocks will drift to some extent, but some hardware can drift as much as one minute for every couple minutes that pass and rapidly get out of sync. NTP is designed to periodically update the system time to account for normal drift. It cannot reasonably correct clocks that drift significantly. This is very uncommon, but there are a few methods that can potentially work around these problems.

The best way to avoid time keeping problems is to use quality hardware that has been tested and does not experience these issues, such as official appliances found in the Netgate Store.

There are four items to check if hardware has significant time keeping problems.

Network Time Protocol

By default, pfSense attempts to synchronize its time using the ntp.org Network Time Protocol (NTP) server pool. This ensures an accurate date and time on the firewall, and will accommodate normal clock drift. If the firewall date and time are incorrect, ensure NTP synchronization is functioning. The most common problem preventing synchronization is the lack of proper DNS configuration on the firewall. If the firewall cannot resolve hostnames, NTP synchronization will fail. The results of synchronization are shown at boot time in the system log, and the status of the NTP clock synchronization can be viewed at **Status > NTP**. The **NTP Status** widget for the Dashboard also offers basic information about the NTP server selected for use by the firewall.

BIOS Updates

We have seen older hardware that ran fine for years on Windows encounter major timekeeping problems once redeployed on FreeBSD (and by consequence, pfSense). The systems were running a BIOS version several revisions out of date. One of the revisions addressed a timekeeping issue that apparently never affected Windows. Applying the BIOS update fixed the problem. The first thing to check is to make sure the hardware in question has the latest available BIOS.

PNP OS settings in BIOS

Other hardware might have time keeping difficulties in FreeBSD and pfSense unless **PNP OS** in the BIOS was set to *No*. If the BIOS does not have a **PNP OS** configuration option, look for an **OS** setting and set it to *Other*.

Disable ACPI

A few BIOS vendors have produced ACPI (Advanced Configuration and Power Interface) implementations which are buggy at best and dangerous at worst. On more than one occasion we have encountered hardware that would not boot or run properly while ACPI support is active.

While ACPI support can be disabled in the BIOS, and in the OS, we do not recommend using hardware that requires such changes.

Adjust Timecounter Hardware Setting

On rare systems, the `kern.timecounter.hardware` sysctl value may need to be changed to correct an inaccurate clock. This is known to be an issue with older versions of VMware such as ESX 5.0 in combination with an amd64-based pfSense or FreeBSD image. That special case was a bug in the hypervisor that is fixed in ESX 5.1 and later.

On these systems the default timecounter will eventually stop the clock from ticking, causing problems with encryption, VPNs, and services in general. On other systems, the clock may skew by large amounts with the wrong timecounter.

To change the timecounter, browse to **System > Advanced**, on the **System Tunables** tab and add an entry to set `kern.timecounter.hardware` to `i8254`

This will make the system use the i8254 timecounter chip, which typically keeps good time but may not be as fast as other methods. The other timecounter choices will be explained later in this section.

If the system keeps time properly after making this change, leave the tunable entry in place to make this change permanent. If the change did not help, remove the tunable or try another value.

Depending on the platform and hardware, there may also be other timecounters to try. For a list of available timecounters found on a firewall, execute the following command:

```
# sysctl kern.timecounter.choice
```

The firewall will print a list of available timecounters and their "quality" as reported by FreeBSD:

```
kern.timecounter.choice: TSC-low(1000) ACPI-safe(850) i8254(0) dummy(-1000000)
```

Try any of those four values for the sysctl `kern.timecounter.hardware` setting. In terms of "quality" in this listing, the larger the number the better, but the actual usability varies from system to system.

TSC A counter on the CPU, but is tied to the clock rate and is not readable by other CPUs. It can be used in bare metal SMP systems but it requires that TSCs on all CPUs be synchronized. It cannot be used reliably on systems with variable-speed CPUs or virtualized system with multiple CPUs.

i8254 A clock chip found in most hardware, which tends to be safe but can have performance drawbacks.

ACPI-safe If it is properly supported by the hardware, this is a good choice because it does not suffer from the performance limitations of i8254, but in practice its accuracy and speed vary widely depending on the implementation.

ACPI-fast A faster implementation of the ACPI timecounter available on hardware that does not suffer from known ACPI issues.

HPET High Precision Event Timer available in some hardware. When available, it is generally considered a good source of accurate time counting.

This and more information on FreeBSD Timecounters can be found in the paper Timecounters: Efficient and precise timekeeping in SMP kernels by Poul-Henning Kamp of the FreeBSD Project, and in the FreeBSD source code.

Adjust the Kernel Timer Frequency

In rare cases adjusting the kernel timer frequency, or `kern.hz` kernel tunable, can help performance or stability. This is especially true on virtualized environments. The default is `1000`, but in some cases `100`, `50`, or even `10` will be a better value depending on the system. When pfSense is installed in VMware, it detects it and automatically sets this tunable to `100`, which should work fine in nearly all cases with VMware products.

To adjust this setting, add a line to `/boot/loader.conf.local` with the new value:

```
kern.hz=100
```

33.56.2 GPS Time Synchronization

To aid in maintaining an accurate clock, GPS time synchronization is also provided by pfSense on supported hardware. Certain serial or USB GPS devices are supported, and in combination with external time servers, they can help keep the clock accurate. For more detail, see *NTPD*.

33.57 Troubleshooting Time Zone Configuration

If the clock is several hours off, but accurate to the minute, it is most likely a time zone setting issue. If using a GMT offset time (e.g. *-0500*), try using a more specific **geographic** time zone such as *America/New_York* instead.

Processes on FreeBSD (and thus pfSense® software) only pick up time changes when they are started. If the firewall has not been rebooted since the last time zone change, doing so will ensure that all running processes are using the correct time zone.

33.58 Troubleshooting Traceroute Output

When *traceroute* is run from LAN to a destination on the Internet, the router itself may be missing from the *traceroute* output depending on the firewall configuration.

This happens on Multi-WAN due to the way that *route-to* and *reply-to* work. policy routing(route-to/reply-to) does not decrease the IP TTL when forwarding packets, so the router does not appear as a hop.

This may also happen with IPsec due to the way IPsec traffic is handled in the kernel. The traffic is not "routed" in a traditional sense.

This behavior may change in future versions of pfSense® software, see here: https://redmine.pfsense.org/issues/932

33.59 Troubleshooting Traffic Shaping

Traffic Shaping/QoS is a tricky topic, and can prove difficult to get right the first time. This section covers several common pitfalls.

33.59.1 Bittorrent traffic not using the P2P queue

Bittorrent is known for not using standard ports. Clients are allowed to declare which port other clients use to reach them, which means chaos for network administrators trying to track the traffic based on port alone. Clients can also choose to encrypt their traffic. Regular shaper rules don't have any way to examine the packets to tell what program the traffic appears to be, so it is forced to rely on ports. This is why it may be a good idea to use the P2P Catchall rule, and/or make rules for each type of desirable traffic and treat the default queue as low priority.

33.59.2 UPnP traffic shaping

Out of the box, traffic allowed in by the UPnP daemon will end up in the default queue. This happens because the rules generated dynamically by the UPnP daemon do not have any knowledge of queues unless UPnP is configured to send traffic into a specific queue.

Depending on what the client devices utilizing UPnP on a network, this may be low priority traffic like Bittorrent, or high priority traffic like game consoles or voice chat programs like Skype.

To configure UPnP to use a specific ALTQ queue:

- Setup ALTQ shaping and decide which queue to use for UPnP & NAT-PMP
- Navigate to **Services > UPnP & NAT-PMP**
- Enter the chosen ALTQ queue name into the **Traffic Shaping** field
- Click **Save**

This trick only works with the ALTQ shaper. At this time, the firewall is not capable of assigning UPnP traffic to a limiter.

33.59.3 ACK queue bandwidth calculations

This is a complex topic and most users gloss over it and guess a sufficiently high value. For more detailed explanations with mathematical formulas, check the Traffic Shaping section of the Netgate forum. There is a sticky post in that board which describes the process in great detail, and there is also a downloadable spreadsheet which can be used to help ease the process.

33.59.4 Why is <x> not properly shaped?

The reason is nearly always one of these choices:

- The traffic matched a different rule than expected
- The traffic did not match any rule

As with other questions in this section, this tends to happen because of rules entered either internally or by other packages that do not have knowledge of queues. Since no queue is specified for a rule, it ends up in the default or root queue, and not shaped.

Working around the limitation may require altering the rules to better match the traffic, or disabling internal rules that are matching the traffic in unexpected ways. Another tactic is to identify all other traffic and then use different shaping options on the default queue.

In rare cases, such as bittorrent, it may be impossible to accurately identify all traffic of a given type. One workaround is to isolate the traffic to one specific device on the network and then match based on that client device address.

33.59.5 WAN connection speed changes

To update the speed of a WAN if it changes, edit the appropriate queues under **Firewall > Traffic Shaper** to reflect the new speed.

The queues that need updating are:

- The root queue for each WAN interface for the upload speed
- The root queue for each LAN interface for the download speed
- qInternet queue for each LAN interface for the download speed

If this firewall has multiple WANs, the LAN root and qInternet queue must use the total download speed of **all** WANs.

Alternately, if the wizard created all of the queues and rules and these have not been changed, then complete the wizard again and update the speed using the wizard.

33.60 Troubleshooting Traffic Shaping Graphs

The RRD for traffic shaping graphs must be reset when a change is made to the traffic shaper settings. The RRD files are in a very specific format and refer to the number and name of the queues as they exist in the shaper configuration. Should this data change, the RRD file data becomes invalid and must be reset.

Therefore any time a traffic shaper setting is changed, the *queue* and *queuedrops* graphs are reset in order to ensure that the RRD schema matches up with the current shaper configuration.

33.61 Troubleshooting Unexpected Reboots

Unexpected reboots are caused by one of two things – hardware problems, or FreeBSD kernel panics. The vast majority of the time, it is hardware problems. Hardware diagnostics should be run before trying anything else.

If the reboot was caused by a kernel panic, the firewall has swap space available, a prompt will be shown when logging back into the web interface asking to *view and submit a crash report*. If there is no kernel panic, then the cause is almost certainly a hardware problem.

For hardware issues, check on:

- Failing power supply
- Flaky electricity in general
- Overheating CPU
- Overheating or bad RAM
- Faulty hard drive/SSD/other storage
- Faulty drive cables
- and many others. . .

If the firewall does panic, and the panic message contains a backtrace that mentions things like memory allocation, mbuf, uma_zalloc_arg, or similar, then it may be crashing due to mbuf exhaustion. See *Tuning and Troubleshooting Network Cards* for information on how to overcome that problem.

On systems without swap space, an automatic crash dump will not happen. The details of the crash and backtrace would need to be viewed on the console if possible.

33.62 Troubleshooting Upgrades

This document describes methods of troubleshooting problems firewalls may encounter when attempting to run a pfSense® upgrade.

33.62.1 Cosmetic Problems Post-Upgrade

If cosmetic problems occur after performing an upgrade, this is nearly always due to stale browser cache entries for CSS, JavaScript, or other files where the browser does not pull the updated version. Force a refresh of the page in the browser (e.g. Shift+Reload or Ctrl+F5) or clear the browser cache to resolve the issues.

33.62.2 Upgrade Log

`pfSense-upgrade` keeps a log of the last update attempt, which may contain additional useful information. This log is located at `/conf/upgrade_log.latest.txt`. Please include the contents of this log with any post or support request when requesting assistance with upgrade problems.

33.62.3 Upgrade not Offered / Library Errors

If the update system does not offer an upgrade to the most recent version, the upgrade will not proceed, or the upgrade process encounters errors with shared libraries, take the following steps:

- Navigate to **System > Updates**
- Set **Branch** to *Previous stable version*
- Wait a few moments for the upgrade check to complete
- Optional: Confirm that the latest version of *pfSense-upgrade* is present using `pkg-static info -x pfSense-upgrade`. For example, it should be at version >= 0.70 for pfSense 2.4.5-p1.

  If the correct version is not present, wait a bit longer and check again as that package may be updating in the background.
- Set **Branch** to *Latest stable version*
- Wait a few moments for the upgrade check to complete

If the upgrade is still not offered, refresh the repository configuration and upgrade script by running the following commands from the console or shell:

```
# pkg-static clean -ay; pkg-static install -fy pkg pfSense-repo pfSense-upgrade
```

If that procedure results in an error, or the upgrade is still not offered, then attempt to update to an intermediate version. For example, to get from Plus 2.4.5-p1 to Plus 21.05 or later, upgrade to Plus 21.02.x before proceeding to later versions if a direct upgrade does not succeed. In these cases, the appropriate version will be visible as a branch to select.

33.62.4 Repository Metadata Version Errors

If `pkg` is unable to update and complains about the repository metadata version, the `pkg` utility may need to be updated manually to version 1.13.x or later.

Example metadata version error:

```
>>> Updating repositories metadata...
Updating pfSense-core repository catalogue...
pkg-static: repository meta /var/db/pkg/pfSense-core.meta has wrong version 2
pkg-static: Repository pfSense-core load error: meta cannot be loaded No error: 0
```

To correct the problem, manually bootstrap `pkg` from an ssh or console shell:

```
# pkg-static bootstrap -f
```

Warning: Do not run the above command from **Diagnostics > Command** as it requires interactive input. If ssh and console shells are unavailable, use this variation instead:

```
env ASSUME_ALWAYS_YES=yes pkg-static bootstrap -f
```

33.62.5 Rewrite Repository Information

In some cases the repository information may need to be rewritten:

- Navigate to **System > Updates**
- Set the **Branch** to *Latest Development Snapshots*
- Wait for the page to refresh
- Set the **Branch** to *Latest stable version*

If the update still does not appear, run the commands above from the console or shell.

33.62.6 CLI Troubleshooting

If the GUI update is not functioning as expected, there are a handful of shell commands that can help gather information or resolve problems.

Force pkg Metadata Update

Often times DNS or connectivity problems will prevent the firewall from finding updates. A quick way to verify this is to force a *pkg* metadata update:

```
# pkg-static update -f
```

This command forces an update and will print errors if problems are found, a few potential errors include:

No address record The firewall cannot resolve the update server hostname. This could be a problem with DNS from the firewall itself, or connectivity from the firewall to the Internet in general, such as a missing or incorrect default route.

No route to host The firewall cannot reach the update server because it cannot find a route there. Most likely, the firewall is missing its default route or the WAN with the default route is down.

Operation timed out The firewall was unable to download a file in a timely manner. This is most likely due to degraded connectivity between the firewall and the update servers. It could also be a routing issue, or a problem with IPv6 on the firewall (See *IPv6 Connectivity Problems*).

Authentication error There is a proxy between the firewall and the update servers and it requires authentication. Move the firewall so it is not behind a proxy, or configure the proxy under **System > Advanced**, **Miscellaneous** tab.

No trusted public keys found The firewall is attempting to update from the wrong repository. Ensure the correct branch is selected as mentioned in *Rewrite Repository Information*. May require a reinstall to resolve. For CE installations, try the following command:

```
# fetch -qo /usr/local/share/pfSense/keys/pkg/trusted/ \
 https://raw.githubusercontent.com/pfsense/pfsense/RELENG_2_4_5/src/usr/local/share/
↪pfSense/keys/pkg/trusted/pkg.pfsense.org.20160406
```

Manual Update Check

To run a manual update check from the CLI:

```
# pfSense-upgrade -d -c
```

When run successfully, this command will print a line stating that a new version is available, and the version number of the available update. Errors displayed during that process are likely to be the same as those covered in *Force pkg Metadata Update*.

33.62.7 packages.netgate.com Has no A/AAAA Record

`pkg` does not use A/AAAA records. It uses service (SRV) records. The update server meta names such as `packages.netgate.com` are not meant to be accessed directly using a browser.

To find the actual update servers, lookup the SRV record for the host:

```
$ host -t srv _https._tcp.packages.netgate.com
_https._tcp.packages.netgate.com has SRV record 10 10 443 files01.netgate.com.
_https._tcp.packages.netgate.com has SRV record 10 10 443 files00.netgate.com.

$ host files01.netgate.com.
files01.netgate.com has address 162.208.119.40
files01.netgate.com has IPv6 address 2607:ee80:10::119:40

$ host files00.netgate.com.
files00.netgate.com has address 162.208.119.41
files00.netgate.com has IPv6 address 2607:ee80:10::119:41
```

Accessing the hosts using their direct hostnames will work with a browser:

```
$ curl --output /dev/null --silent --head --fail \
 "https://files00.netgate.com/pfSense_v2_4_5_amd64-core/meta.txz"
$ echo $?
0
```

33.62.8 IPv6 Connectivity Problems

If IPv6 is configured on the firewall, the pfSense software will prefer to use it when performing an update. There are cases when a firewall may have broken IPv6 connectivity, however, that contribute to problems updating. This could manifest as a timeout or routing error when upgrading.

Typically the operating system will attempt to fall back to IPv4, but the extra time this takes could also lead to a timeout.

The firewall can be configured to prefer IPv4 to eliminate this as a potential cause. See *Controlling IPv6 Preference for traffic from the firewall itself* for details.

Alternately, from ssh or a console shell, force the upgrade to use IPv4 manually:

```
# pfSense-upgrade -4
```

33.62.9 Segmentation Fault in pkg

Certain cryptographic hardware can have a software-induced race condition which leads to a problematic state. In this state, `pkg` will crash with a segmentation fault:

```
1085486128:error:14099044:SSL routines:ssl3_send_client_verify:internal error:
Child process pid=30149 terminated abnormally: Segmentation fault
```

In this case, the device must be powered off and back on to recover. A warm reboot is not sufficient to reset the hardware.

- Navigate to **Diagnostics > Halt System**
- Click  **Halt**
- Wait for the device to shut down. Monitor the console to ensure that the shutdown completes.
- Unplug the power adapter
- Plug the power adapter back in

33.62.10 Forced pkg Reinstall

Forcing a reinstallation of all packages may resolve problems that otherwise may require a full reinstall. This is not ideal, as a clean install is more likely to have a positive result, but that is not always an option in every situation (e.g. remote install with no console access).

To forcefully reinstall all packages, take the following steps:

- Make a backup
- Clean the repository and forcefully reinstall pkg, repo data, and the upgrade script:

```
# pkg-static clean -ay; pkg-static install -fy pkg pfSense-repo pfSense-upgrade
```

- Force a reinstall of everything:

```
# pkg-static upgrade -f
```

- Review the list of changes and enter `y` to proceed
- Manually reboot the firewall

33.62.11 Last Resort

If nothing else works then a reinstall will eliminate any possibility of problems related to the upgrade itself.

pfSense software supports multiple options to easily restore the configuration. The fastest method is **Recover config.xml** as discussed in *Automatically Restore Configuration During Installation*. Using that method, the pfSense installation can pick up the existing configuration from the existing install and use it, eliminating the need for any manual restore process. The firewall will boot up after installation with the old settings and reinstall packages as needed.

33.63 Troubleshooting Website Access

If some sites will load, but other sites will not, there are a few possible causes.

1. Check all of the items listed on *Connectivity Troubleshooting* before proceeding
2. Ensure the WAN gateway is reachable and set to the proper address
3. Ensure the subnet mask on the WAN interface of the firewall is correct
4. Ensure the subnet mask on the client stations and on every interface (and VPN) in the pfSense® configuration is correct
5. Ensure the WAN MTU is properly set (See here to determine the MTU), use MSS to lower the MTU if necessary
6. Use *traceroute* to determine where the traffic stops. It may be an upstream connectivity issue and not the pfSense firewall or ISP.
7. *Disable hardware checksums* and see if the problem disappears
8. Disable any proxy package such as squid that is in use (if any)
9. Check **Clear invalid DF bits instead of dropping the packets** on **System > Advanced**, **Firewall/NAT** tab.
10. Check **Disable Firewall Scrub** on **System > Advanced**, **Firewall/NAT** tab.

33.64 Troubleshooting Wireless Connections

When it comes to wireless, there are a lot of things that can go wrong. From faulty hardware connections to radio interference to incompatible software/drivers, or simple settings mismatches, anything is possible, and it can be a challenge to make it all work on the first try. This section will cover some of the more common problems that have been encountered by pfSense® users and developers.

33.64.1 Check the Antenna

Before spending any time diagnosing an issue, double and triple check the antenna connection. If it is a screw-on type, ensure it is fully tightened. For mini-PCI cards, ensure the pigtail connectors are properly connected and snapped in place. Pigtails on mini-PCI cards are fragile and easy to break. After disconnecting and reconnecting pigtails a few times, they may need to be replaced.

33.64.2 Check Wireless Status

The status of connected wireless clients and nearby access points can be viewed by navigating to **Status > Wireless**. This menu option only appears when a wireless interface is present and enabled.

On this page, click **Rescan** and then refresh the page after 10 seconds to see other nearby access points. If they are on the same or a nearby channel, there could be interference.

In the list of associated clients, several flags are listed that explain the capabilities of the connected client. For example if the client has an “H” flag, this indicates High Throughput used by 802.11n. If a client is connected without that flag, they may be using an older lower standard. A full list of wireless flags is contained in *Wireless Status*, including access point capability descriptions.

33.64.3 Try with multiple clients or wireless cards

To eliminate a possible incompatibility between wireless functions on pfSense and a wireless client, be sure to try it with multiple devices or cards first. If the same problem is repeatable with several different makes and models, it is more likely to be a problem with the configuration or related hardware than the client device.

33.64.4 Signal Strength is Low

If the signal is weak even when nearby the access point antenna, check the antenna again. For mini-PCI or mini-PCIe cards, if only one pigtail in use and there are two internal connectors, try hooking the pigtail up to the other internal connector on the card. Also try changing the **Channel** or adjusting the **Transmit Power**, or the **Antenna Settings** on the wireless interface configuration. For mini-PCI and mini-PCIe cards, check for broken ends on the fragile pigtail connectors where they plug into the card. If the **Regulatory Domain** settings have not been configured, set them before testing again.

33.64.5 Stuck Beacon Errors

If a "Stuck Beacon" error is found in the system or wireless log, it is usually an indication that the chosen wireless channel is too noisy:

```
kernel: ath0: stuck beacon; resetting (bmiss count 4)
```

The sensitivity of this behavior can be tuned by adding a System Tunable entry for `hw.ath.bstuck` with a value of `8` or higher.

If the errors persist, use a WiFi survey app or program to determine an open or less-used channel to use instead of the current channel.

33.64.6 Interface Unavailable for Assignment

If a wireless interface does not appear in the list of interfaces **Interfaces > Assignments** there are two possibile issues:

If the wireless card is supported, a wireless instance must first be created as described in *Creating and Managing Wireless Instances*. Once the instance is created, it will be available for assignment.

If the wireless card is not supported, it will not be available for selection as a parent interface when creating a wireless instance.

33.65 General

- *Troubleshooting Captive Portal*
- *Troubleshooting Offline DHCP Leases*
- *Troubleshooting DHCPv6 Client XID Mismatches*
- *Troubleshooting FTP Connections*
- *Troubleshooting ARP Move Log Messages*
- *Troubleshooting Clock Issues*
- *Troubleshooting Time Zone Configuration*

33.66 Authentication / User Manager

- *Troubleshooting Authentication*
- *Troubleshooting Access when Locked Out of the Firewall*
- *Troubleshooting "login on console as root" Log Messages*

33.67 Connectivity / Networking

- *Troubleshooting Network Connectivity*
- *Troubleshooting GUI Connectivity*
- *Troubleshooting "No buffer space available" Errors*
- *Troubleshooting Website Access*
- *Troubleshooting Low Interface Throughput*
- *Troubleshooting Lost Traffic or Disappearing Packets*
- *Troubleshooting Traceroute Output*

33.68 DNS

- *Troubleshooting DNS Resolution Issues*
- *Troubleshooting the DNS Cache*
- *Troubleshooting the DNS Forwarder*
- *Troubleshooting Thread Errors with Hostnames in Aliases*

33.69 Hardware

- *Troubleshooting Boot Issues*
- *Troubleshooting in Single User Mode*
- *Troubleshooting DMA and LBA Errors*
- *Troubleshooting Disk and Filesystem Issues*
- *Troubleshooting Full Filesystem or Inode Errors*
- *Troubleshooting High CPU Load*
- *Troubleshooting "promiscuous mode enabled" Log Messages*
- *Troubleshooting Unexpected Reboots*
- *Troubleshooting Wireless Connections*
- *Troubleshooting Hardware Shutdown and Power Off*

33.70 High Availability

- *Troubleshooting High Availability*
- *Troubleshooting High Availability Clusters in Virtual Environments*
- *Troubleshooting High Availability DHCP Failover*
- *Troubleshooting VPN Connectivity to a High Availability Secondary Node*

33.71 Installation / Upgrades

- *Troubleshooting Installation Issues*
- *Troubleshooting Upgrades*
- *Troubleshooting a Broken pkg Database*

33.72 Rules/NAT

- *Troubleshooting Firewall Rules*
- *Troubleshooting Bogon Network List Updates*
- *Troubleshooting Blocked Log Entries for Legitimate Connection Packets*
- *Troubleshooting NAT*
- *Troubleshooting 1:1 NAT*
- *Troubleshooting NAT Port Forwards*
- *Troubleshooting NAT Reflection*
- *Troubleshooting Traffic Shaping*
- *Troubleshooting Traffic Shaping Graphs*

33.73 Routing / Multi-WAN

- *Troubleshooting Routes*
- *Troubleshooting Asymmetric Routing*
- *Troubleshooting Multi-WAN*
- *Troubleshooting Gateway Monitoring*

33.74 VPN

- *Troubleshooting IPsec VPNs*
- *Troubleshooting Duplicate IPsec SA Entries*
- *Troubleshooting OpenVPN*
- *Troubleshooting OpenVPN Remote Access Client IP Address Assignments*
- *Troubleshooting Windows OpenVPN Client Connectivity*
- *Troubleshooting Windows/SMB Share Access from OpenVPN Clients*
- *Troubleshooting OpenVPN Internal Routing (iroute)*
- *Troubleshooting OpenVPN Push Routes*
- *Troubleshooting L2TP*
- *Troubleshooting Cisco VPN Pass Through*

33.75 Packages

- *Troubleshooting the HAProxy Package*
- *Troubleshooting Snort Rule Updates*
- *Troubleshooting the Squid Package*

CHAPTER

THIRTYFOUR

PFSENSE CONFIGURATION RECIPES

The recipes in this section walk administrators through configuring various aspects of pfSense® software.

34.1 Authenticating Users with Google Cloud Identity

Google Cloud Identity LDAP service can be used to authenticate users on pfSense® software installations.

The method varies depending on the version of pfSense software installed on the firewall. This is due to the fact that Google Cloud Identity requires a client certificate to make a secure LDAP connection.

- Firewalls running pfSense factory software version 2.4.4-RELEASE-p1 or later can use a client certificate directly on LDAP authentication sources.
- Firewalls running pfSense CE or pfSense factory software version 2.4.4-RELEASE require the stunnel package to make a secure LDAP connection.

Configuring a firewall running pfSense software to use G Suite LDAP authentication requires a number of steps, all of which are covered in this document.

34.1.1 Configure the LDAP Application on the G Suite admin portal

Follow the instructions from Google for configuring and enabling the G Suite LDAP application.

> **Warning:** **Follow these directions exactly**. No special provisions are required for pfSense, but please note that the LDAP application credentials (username and password) **are required**.

34.1.2 Download the certificate, key, username and password

Download the certificate, key, username and password from G Suite to a local directory on a workstation.

34.1.3 Import the certificate and key

From the web interface of a firewall running pfSense:

- Navigate to **System > Cert manager**, **Certificates** tab
- Click  **Add/Sign** to display the certificate import interface
- Change **Method** to *Import an existing certificate*
- Enter a **Descriptive name**, such as `G Suite LDAP`
- Copy and paste the contents of the downloaded certificate into the **Certificate data** box
- Copy and paste the contents of the downloaded key into the **Private Key data** box
- Click **Save**

The certificate is now available for use by the firewall.

The next step depends on the version of pfSense software installed on the firewall.

For pfSense CE or pfSense factory software version 2.4.4-RELEASE, the stunnel package is necessary to make a secure LDAP connection. For these environments, proceed to *Install the stunnel pfSense package (CE or 2.4.4-RELEASE)*.

For users of pfSense factory software version 2.4.4-RELEASE-p1 or later, LDAP authentication sources can use a client certificate directly. Skip ahead to *Configure LDAP authentication on pfSense*.

34.1.4 Install the stunnel pfSense package (CE or 2.4.4-RELEASE)

From the web interface on pfSense:

- Navigate to **System > Package manager**, **Installed Packages** tab
- Check the list for **stunnel** and if it is listed as installed
- If the package is installed and up-to-date, with a version of **5.37** or later, no action is required
- If the package is installed but out of date
  - Update the package by clicking  for the **stunnel** entry
  - Click  **Confirm** to confirm the package update
- If **stunnel** is not installed
  - Navigate to the **Available packages** tab
  - Locate the **stunnel** package in the list, or use the search bar
  - Click  **Install** for the **stunnel** package entry
  - Click  **Confirm** to confirm the package installation

34.1.5 Configure the stunnel package (CE or 2.4.4-RELEASE)

From the web interface on pfSense:

- Navigate to **Services > STunnel**
- Click  **Add** to create a new profile
- Enter a **Description** for this connection, such as `G Suite`
- Check **Client Mode**
- Set **Listen on IP** to `127.0.0.1`
- Set **Listen on port** to `1636`
- Set the **Certificate** to the entry imported previously, in this case *G Suite LDAP*
- Set **Redirects to IP** to `ldap.google.com`
- Set **Redirects to port** to `636`
- Click **Save**

34.1.6 Configure LDAP authentication on pfSense

From the web interface on pfSense:

- Select **System > User manager**, **Authentication servers** tab
- Click  **Add** to create a new entry
- Enter a **Descriptive name** for this LDAP server, such as `G Suite`
- Set **Type** to *LDAP*
- The server settings depend on the pfSense software version installed on the firewall:
  - For pfSense Factory version 2.4.4-RELEASE-p1 or later:
    - Set the **Hostname or IP address** to `ldap.google.com`
    - Set **Port value** to `636`
    - Set **Transport** to *SSL - Encrypted*
    - Set **Peer Certificate Authority** to *Global Root CA List*
    - Set *Client Certificate* to the entry imported previously, in this case *G Suite LDAP*
  - For pfSense CE or factory version 2.4.4-RELEASE using stunnel:
    - Set the **Hostname or IP address** to `127.0.0.1`
    - Set **Port value** to `1636`
    - Set **Transport** to *TCP-Standard*
- Set **Protocol version** to *3*
- Set **Server timeout** = `25`
- Set **Search scope** to *Entire tree*

The next few settings are **UNIQUE TO THE DOMAIN**. For this example, assume that is `example.com`.

> **Warning:** Substitute the actual domain when entering these values!

- Set **Base DN** to the domain name in DN format, for example `dc=example,dc=com`
- Set **Authentication containers** to the **Base DN** prepended by the `Users` organizational unit, for example: `ou=Users,dc=example,dc=com`
- **Uncheck** the **Bind anonymous** box to show the **Bind Credentials** fields
- Set **Bind credentials** to the G Suite LDAP username and password that were created with the certificate and key

The remaining attributes are not specific to the domain, or are defaults

- Set **User naming attribute** to `uid`
- Set **Group naming attribute** to `cn`
- Set **Group member attribute** to `memberOf`

34.1.7 Create a Group

Using a remote authentication server to manage administrative logins to services on pfSense requires a matching group to be present on both the authentication source server and on the firewall. The existing `admins` group could be used, but since the name is so general it may conflict with other desired permissions in G Suite.

This example uses a new group called `fwadmins`.

First, create the `fwadmins` group in G Suite and assign users to the group. The exact details will vary based on the domain and its organization.

Next, create a group on the firewall running pfSense software. This does not require local users, only a group entry. The group entry must have appropriate permissions.

To create the group on pfSense:

- Navigate to **System > User Manager**, **Groups** tab
- Click  **Add** to make a new group
- Enter the **Group name**, in this example: `fwadmins`
- Set the **Scope** to *Remote*
- Enter a **Description**, such as **Remote Firewall Administrators**
- Click **Save**

Now the group needs privileges:

- Click  on the row for the newly created group
- Click  **Add** in the **Assigned Privileges** section
- Select the desired permissions for the group, for example: `WebCfg - All pages`

> **Warning:** **Do not select every item** in this list! Doing so will also select the `User - Config: Deny Config Write` privilege which will prevent users in this group from making changes to the firewall configuration.

- Click **Save** to store the privileges

34.1.8 Test G Suite Authentication

With the complete configuration described above, it is now possible to authenticate against Google G Suite LDAP. First, test the authentication to ensure it is working properly.

- Navigate to **Diagnostics > Authentication**
- Set the **Authentication server** to the name used for the LDAP Server entry, such as *G Suite*
- Enter a known username and password on the domain that G Suite controls

> **Note:** By default only the username part of the login is checked against the configured LDAP base DN. If a username is submitted with a domain part, for example `user@example.com`, the `@example.com` part is ignored.

- Click  **Test**

The user should show as authenticating successfully, and if the user entered is a member of the `fwadmins` group in G Suite, that should also be reflected in the test output.

If the test succeeds, the service is ready for use. pfSense can use it as an authentication source for the GUI, for VPNs, or anywhere the user manager authentication servers work.

If the test fails, check the main system log for error messages from LDAP. Start from the beginning of this document and compare all settings between this document, G Suite, and pfSense. Most common problems are with parameters being input incorrectly, such as selecting the wrong certificate, using an incorrect LDAP attribute name, or not using correct bind credentials.

34.1.9 Use G Suite for pfSense Administrative Logins

If all is well and the user authenticated as expected:

- Navigate to **System > User manager**, **Settings**
- Set the **Authentication server** to *G Suite*
- Click **Save**

After saving, firewall users will be authenticated against Google Cloud Identity.

> **Note:** pfSense will automatically fall back to local authentication if it cannot authenticate using the chosen LDAP server.

34.2 Configuring BIND as an RFC 2136 Dynamic DNS Server

If the DNS for a domain is directly controlled on a BIND server, RFC 2136 Dynamic DNS support can be setup for use by pfSense®. This section shows how to configure BIND to support this feature.

The exact location of the configuration directory for BIND will vary by operating system. It could be in `/usr/local/etc/namedb/`, `/etc/namedb/`, or elsewhere.

See also:

See *Configuring RFC 2136 Dynamic DNS updates* for more information on RFC 2136 Dynamic DNS.

34.2.1 Configure the BIND Server

On the server in `named.conf`, add the following block:

```
include "/etc/namedb/dns.keys.conf";
zone "dyn.example.com" {
     type master;
     file "dynamic/dyn.example.com";
     update-policy { grant *.dyn.example.com. self dyn.example.com. A AAAA; };
};
```

Then create the initial zone file. BIND requires read/write access to this file and the directory in which it resides so that the zone and its journal may be updated.

Warning: BIND will rewrite this zone file, which is why a subdomain is used in the example.

From there, create the zone file for the new dynamic zone, `dynamic/dyn.example.com`

```
$ORIGIN      .
$TTL 30      ; 30 seconds
dyn.example.com             IN SOA  ns.example.com. hostmaster.example.com. (
                            2016062801 ; serial
                            3600       ; refresh (1 hour)
                            600        ; retry (10 minutes)
                            2600       ; expire (43 minutes 20 seconds)
                            30         ; minimum (30 seconds)
                            )
                      NS    ns.example.com.
                      NS    ns2.example.com.
```

Reload the named service using `rndc reload` or a similar command, and then if any slave name servers are in place, add a zone to those servers as well:

```
zone "dyn.example.com" {
     type slave;
     file "dynamic/dyn.example.com";
     masters{ 192.0.2.5; };
};
```

For BIND 9.16+ create an entry using `tsig-keygen`:

```
# tsig-keygen -a hmac-md5 myhost.dyn.example.com
key myhost.dyn.example.com. {
     algorithm hmac-md5;
     secret "/0/4bxF9A08n/zke/vANyQ==";
};
```

Add that key to `dns.keys.conf` manually or by redirecting command output:

```
# tsig-keygen -a hmac-md5 myhost.dyn.example.com >> /etc/namedb/dns.keys.conf
```

For BIND version < 9.16, follow the next steps.

On the master name server, make the keys directory:

```
# mkdir -p /etc/namedb/keys
```

And now generate a host key. The second line is the output of the command, *not* part of the command itself.

```
# /usr/sbin/dnssec-keygen -K /etc/namedb/keys -a HMAC-MD5 -b 128 -n HOST myhost.dyn.
↪example.com.
Kmyhost.dyn.example.com.+157+32768
```

The output `Kmyhost.dyn.example.com.+157+32768` is the first part of the filename for the key, it will append `.private` to one file and `.key` to another. Both contain the same data in different formats.

Now read the key from the new key file:

```
# /usr/bin/grep ^Key: /etc/namedb/keys/Kmyhost.dyn.example.com.+157+32768.private | /
↪usr/bin/awk '{ print $2; }'
/0/4bxF9A08n/zke/vANyQ==
```

And then add that key to `dns.keys.conf`:

```
key myhost.dyn.example.com. {
     algorithm hmac-md5;
     secret "/0/4bxF9A08n/zke/vANyQ==";
};
```

This can be automated with a simple script, `make-ddns-host.sh`:

```
#!/bin/sh
KEY_NAME=${1}
KEY_DIR=/etc/namedb/keys
KEYS_CONFIG=/etc/namedb/dns.keys.conf
/bin/mkdir -p ${KEY_DIR}
cd ${KEY_DIR}
KEY_FILE_NAME=`/usr/sbin/dnssec-keygen -K ${KEY_DIR} -a HMAC-MD5 -b 128 -n HOST ${KEY_
↪NAME}.`
KEY_TEXT=`/usr/bin/grep "^Key:" ${KEY_DIR}/${KEY_FILE_NAME}.private | /usr/bin/awk '{␣
↪print $2; }'`
echo "key ${KEY_NAME}. {" >> ${KEYS_CONFIG}
echo "        algorithm hmac-md5;" >> ${KEYS_CONFIG}
echo "        secret \"${KEY_TEXT}\";" >> ${KEYS_CONFIG}
echo "};" >> ${KEYS_CONFIG}
echo "Key for ${KEY_NAME} is: ${KEY_TEXT}"
```

After making the file, make it executable:

```
# chmod u+x make-ddns-host.sh
```

To use the script:

```
# ./make-ddns-host.sh mynewhost.dyn.example.com
# rndc reload
```

34.2.2 Configuring a Client in pfSense

To add a DynDNS entry in the pfSense GUI:

- Navigate to **Services > Dynamic DNS**, **RFC 2136** tab
- Click  **Add** to create a new entry with the following settings:

  Enable Checked

  Interface *WAN*

  Hostname The fully qualified hostname, e.g. `xxxxx.dyn.example.com`

  TTL `30`

  Key Name The fully qualified hostname again, exactly: `xxxxx.dyn.example.com`

  Key Type **Host**

  Key Secret key for this hostname

  Server The IP address or hostname of the BIND server

  Protocol Unchecked

  Description `My DynDNS Entry`

- Click **Save**

Assuming the firewall has connectivity to the name server, and there are no other access policies that would prevent the update, RFC 2136 DynDNS service is now working. If the update does not work, check the BIND log and the system log on pfSense.

34.3 Blocking Web Sites

There are several options for blocking websites with pfSense® software, some of which are described on this article. It's not an exact science, but these solutions typically function well enough for a majority of use cases.

See also:

The pfBlocker package (*pfBlocker-NG Package*) offers mechanisms which can be useful in this area, such as DNSBL, geographic IP address blocking, and automation of AS lookups.

34.3.1 Using DNS

If the built in DNS Resolver or Forwarder are active an override can be entered there to resolve the unwanted website to an invalid IP address such as `127.0.0.1`.

With the DNS Resolver, additional methods are possible via custom options.

This first example will prevent any host under the given zone from being resolved by clients:

```
server:
local-zone: "movie.edu" static
```

When the firewall enforces DNS resolution in this way, the firewall must also force clients to resolve DNS using the firewall. Otherwise, clients could bypass the restrictions by using alternate DNS servers. See *Redirecting Client DNS Requests* for details.

34.3.2 Using Firewall Rules

If a website rarely changes IP addresses, access to it can be blocked using an alias containing its IP addresses and then using this alias in firewall rules.

Warning: This is not a feasible solution for sites that return low TTLs and spread the load across many servers and/or datacenters, such as Google and similar large sites. Most small to mid sized websites can be effectively blocked using this method as they rarely change IP addresses.

A hostname can also be inside a network alias. The hostname will be resolved periodically and updated as needed. This is more effective than manually looking up the IP addresses, but will still fall short if the site returns DNS records in a way that changes rapidly or randomizes results from a pool of servers on each query, which is common for large sites.

Another option is finding all of a site's IP subnet allocations, creating an alias with those networks, and blocking traffic to those destinations. This is especially useful with sites such as Facebook that spread large amounts of IP space, but are constrained within a few net blocks. Using regional registry sites such as ARIN can help track down those networks. For example, all of the networks used by Facebook in the region covered by ARIN can be found at http://whois.arin.net/rest/org/THEFA-3.html under "Related Networks". Companies may have other addresses in different regions, so check other regional sites as well, such as RIPE, APNIC, etc.

As an alternative to looking up the IP blocks manually, locate the target company's BGP Autonomous System (AS) number by doing a `whois` lookup on one of their IP addresses, then use that to find all of their allocations. For example, Facebook's AS number is `AS32934`:

```
# whois -h whois.radb.net -- '-i origin AS32934' | awk '/^route:/ {print $2;}' | sort␣
↪| uniq
```

Copy the results of that command into a new alias and it will cover all of their currently allocated networks. Check the results periodically for updates.

34.3.3 Using a Proxy

If web traffic flows through a proxy server, that proxy server can likely be used to prevent access to such sites. For example, Squid has an add-on called SquidGuard which allows for blocking web sites by URL or other similar criteria. There is a very brief introduction to Squid and SquidGuard to be found in *A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid*.

In modern environments this is not effective as it works best on HTTP, and not HTTPS. HTTPS can sometimes be filtered via peek/splice to inspect SNI and similar aspects of connections, but even that fails with modern security practices like encrypted SNI. A non-transparent proxy setup is more likely to work correctly, but is more complicated to setup and maintain.

34.3.4 Prevent Bypassing Restrictions

With any of the above methods, there are many ways to get around the defined blocks. The easiest and likely most prevalent is using any number of proxy websites. Finding and blocking all of these individually and keeping the list up to date is impossible. The best way to ensure these sites are not accessible is using an external proxy or content filtering capable of blocking by category.

To further maintain control, use a restrictive egress ruleset and only allow traffic out to specific services and/or hosts. For example, only allow DNS access to the firewall or the DNS servers specifically used for LAN clients (*Redirecting Client DNS Requests*). Also, if a proxy is in use on the network, make sure to disallow direct access to HTTP and HTTPS through the firewall and only allow traffic to and/or from the proxy server.

34.4 Blocking External Client DNS Queries

This procedure configures the firewall to block DNS requests to servers outside the local network. With no other accessible DNS servers, clients are forced to send DNS requests to the DNS Resolver or DNS Forwarder on pfSense® software for resolution.

Note: Blocking is effective but does not gracefully handle the situation. Clients must manually adjust their configuration to use the firewall for DNS. Redirecting DNS requests to the firewall is a more seamless solution. See *Redirecting Client DNS Requests* for details.

- Navigate to **Firewall > Rules**, **LAN** tab
- Create the block rule as the first rule in the list:
  - Click  **Add** to create a new rule at the top of the list
  - Fill in the following fields on the rule:
    - **Action**: *Reject*
    - **Interface**: *LAN*
    - **Protocol**: *TCP/UDP*
    - **Destination**: *Any*
    - **Destination Port Range**: *DNS (53)*
    - **Description**: `Block DNS to Everything Else`
- Create the pass rule to allow DNS to the firewall, above the block rule:

  - Click  **Add** to create a new rule at the top of the list
  - Fill in the following fields on the rule:
    * **Action**: *Pass*
    * **Interface**: *LAN*
    * **Protocol**: *TCP/UDP*
    * **Destination**: *LAN Address*
    * **Destination Port Range**: *DNS (53)*
    * **Description**: `Pass DNS to the Firewall`

- Click  **Apply Changes** to reload the ruleset

When complete, there will be two rule entries that look like the following picture:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✔ 0 /0 B	IPv4 TCP/UDP	*	*	LAN address	53 (DNS)	*	none		Pass DNS to the Firewall	
☐	0 /0 B	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none		Block DNS to Everything Else	

Certain local PCs could be allowed to use other DNS servers by placing a pass rule for them above the block rule.

34.4.1 DNS over TLS

Another concern is that clients could use DNS over TLS to resolve hosts. DNS over TLS sends DNS requests over an encrypted channel on an alternate port, `853`.

This traffic can be blocked with a firewall rule for port `853` using the same procedure used for `53`. Though if the firewall will not be providing DNS over TLS service to clients, do not add the pass rule.

34.4.2 DNS over HTTPS

Similar to DNS over TLS, clients may also use DNS over HTTPS (DoH). This is harder to block as it uses port `443`. Blocking port `443` on common public DNS servers may help (e.g. `1.1.1.1`, `8.8.8.8`).

Some browsers automatically attempt to use DNS over HTTPS because they believe it to be more secure and better for privacy, though that is not always the case. Each browser may have its own methods of disabling this feature, though in the case of Firefox it uses a "canary" domain by default. If this domain name cannot be resolved by the browser, the browser disables DNS over HTTPS.

To prevent Firefox from using DNS over HTTPS, add the following to the DNS Resolver custom options:

```
server:
local-zone: "use-application-dns.net" always_nxdomain
```

34.5 Configuring DNS over TLS

Several popular public DNS providers provide encrypted DNS service using DNS over TLS. This prevents intermediate parties from viewing the content of DNS queries and can also assure that DNS is being provided by the expected DNS servers.

34.5.1 Requirements

This feature is only supported by the *DNS Resolver*. If the firewall is currently using the *DNS Forwarder*, convert to the DNS Resolver before starting this procedure.

Pick a DNS over TLS upstream provider, such as a private upstream DNS server or a public server like Cloudflare, Quad9, or Google public DNS. Note the addresses of the servers and their associated hostnames.

34.5.2 Configure DNS Servers

First, configure the DNS servers on the firewall.

Warning: When the firewall uses DNS over TLS, every DNS server used by the firewall must support DNS over TLS.

- Navigate to **System > General**
- Locate the **DNS Server Settings** Section
- Add or replace entries in the **DNS Servers** section such that only the chosen DNS over TLS servers are in the list.

  Address IP address of an upstream DNS Server providing DNS over TLS service (e.g. `1.1.1.1`).

  Hostname Hostname of the same upstream DNS Server in the **Address** field, used for TLS certificate validation (e.g. `cloudflare-dns.com`).

  Warning: The hostname is technically optional but dangerous to omit. The DNS Resolver must have the hostname to validate that the correct server is providing a given response. The response is still encrypted without the hostname, but the DNS Resolver has no way to validate the response to determine if the query was intercepted and answered by a third party server (Man-in-the-Middle attack).

- Click  **Add DNS Server** and repeat the previous step as needed for each available DNS server.
- Uncheck **Allow DNS server list to be overridden by DHCP/PPP on WAN** as this may add DNS servers to the configuration which do not support DNS over TLS.
- Click **Save**

Use *Example DNS Server list for DNS over TLS from Cloudflare* as a reference for the settings on the page.

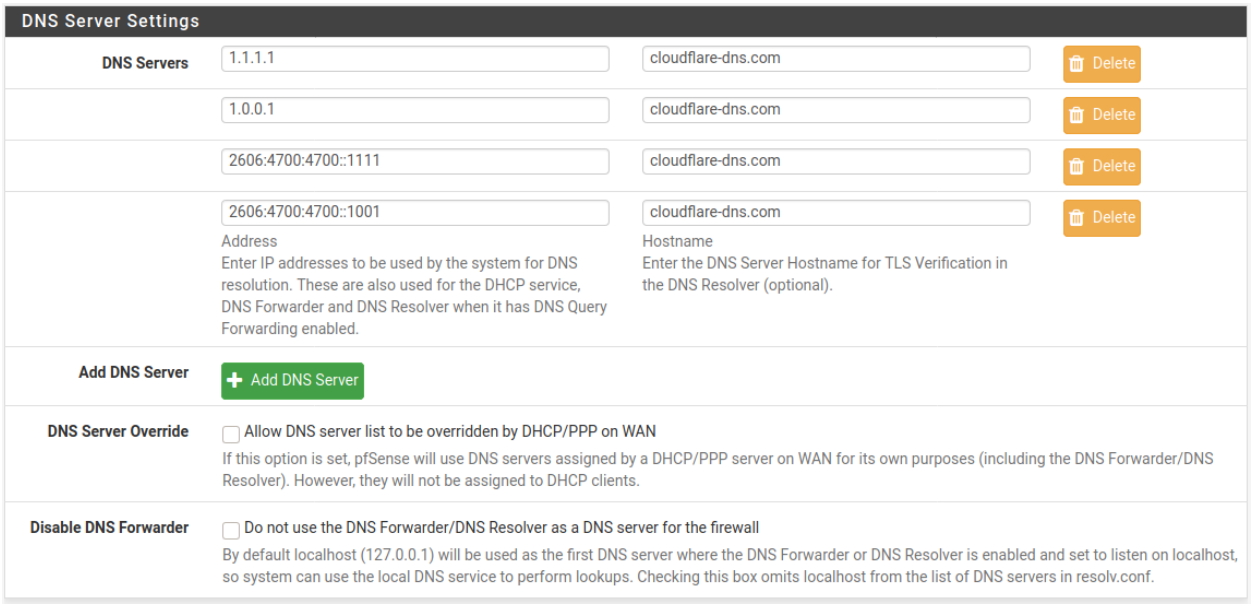


Fig. 1: Example DNS Server list for DNS over TLS from Cloudflare

34.5.3 Enable DNS over TLS for Forwarded Queries

Next, configure the DNS Resolver to use DNS over TLS for outgoing queries.

- Navigate to **Services > DNS Resolver**
- Uncheck **Enable DNSSEC Support**

> **Note:** DNSSEC is not generally compatible with forwarding mode, with or without DNS over TLS.

- Check **Enable Forwarding Mode**
- Check **Use SSL/TLS for outgoing DNS Queries to Forwarding Servers**
- Click **Save**
- Click **Apply Changes**

Use *Example DNS Resolver configuration for outgoing DNS over TLS* as a reference for the settings on the page.

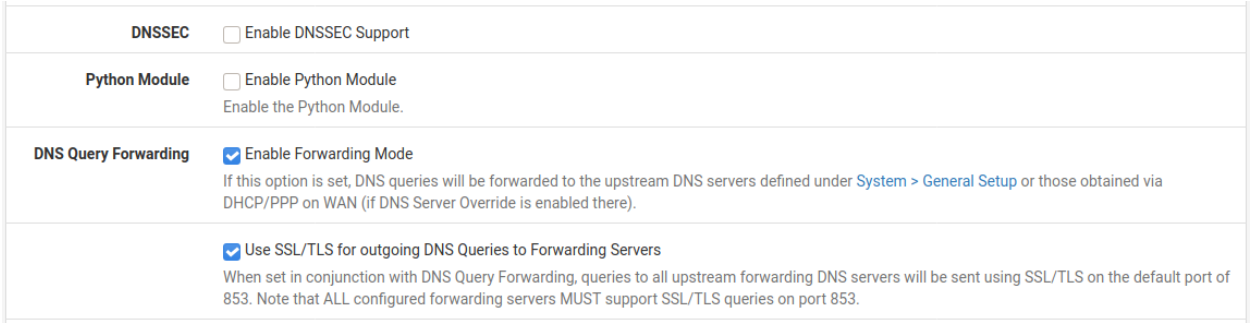


Fig. 2: Example DNS Resolver configuration for outgoing DNS over TLS

The DNS Resolver will now send queries to all upstream forwarding DNS servers using SSL/TLS on the default port of 853.

34.5.4 Testing DNS over TLS

There are several ways to validate that outbound queries are using DNS over TLS.

- Test via Diagnostics > DNS Lookup (*DNS Lookup*) and ensure the result from `127.0.0.1` is correct.
- Check for states using port `853` going to the DNS servers in the configuration (*Firewall States*) like those in *Example State Table contents for DNS over TLS queries*.
- Packet capture port `853` (*Packet Capturing*) and inspect the capture in Wireshark. The contents of the query are not visible, but the TLS exchange is, and any TLS errors in negotiation should be visible in the capture.

States

Interface	Protocol	Source (Original Source) -> Destination (Original Destination)	State	Packets	Bytes
WAN	tcp	198.51.100.23:51344 -> 1.0.0.1:853	FIN_WAIT_2:FIN_WAIT_2	11 / 9	952 B / 4 KiB
WAN	tcp	198.51.100.23:31300 -> 1.1.1.1:853	FIN_WAIT_2:FIN_WAIT_2	11 / 9	947 B / 4 KiB

Fig. 3: Example State Table contents for DNS over TLS queries

34.5.5 Enable DNS over TLS Server (optional)

The DNS Resolver can also act as a DNS over TLS server, though it does not affect outbound/forwarded queries, so this section is optional.

Only enable this feature if *local clients* must talk to the *DNS Resolver* using DNS over TLS queries.

- Navigate to **Services > DNS Resolver**
- Check **Respond to incoming SSL/TLS queries from local clients**
- Select a valid server certificate in **SSL/TLS Certificate**

> **Note:** Clients may reject this certificate if it is self-signed, consider using a certificate from *ACME*.

- Leave **SSL/TLS Listen Port** at the default (empty or `853`)
- Click **Save**
- Click **Apply Changes**

Use *Example DNS Resolver configuration for acting as a DNS over TLS Server* as a reference for the settings on the page.

Now the DNS Resolver will listen for DNS over TLS queries from local clients on port `853`.

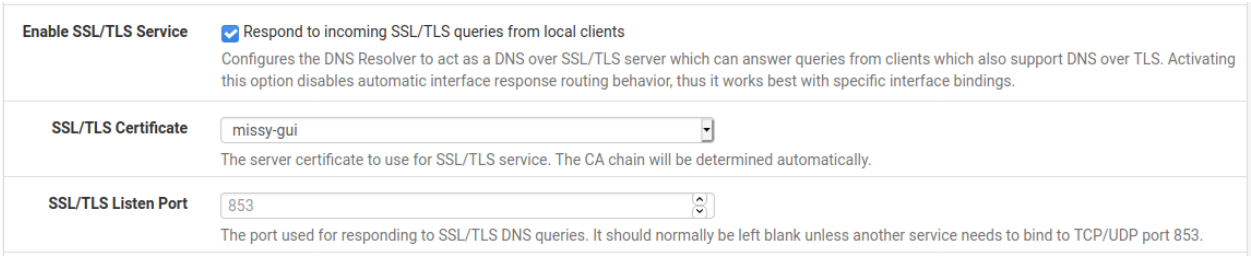


Fig. 4: Example DNS Resolver configuration for acting as a DNS over TLS Server

34.5.6 Caveats

Clients can make their own connections to DNS over TLS servers, so block them on TCP/UDP ports `53` and `853` to ensure they only query the DNS Resolver (*Blocking External Client DNS Queries*).

Redirecting DNS over TLS queries to the DNS Resolver may or may not work, depending on the clients. Setup the *DNS over TLS server* and add port forward redirects for TCP/UDP ports `53` and `853` to redirect DNS queries to the firewall (*Redirecting Client DNS Requests*).

Note: Though clients may reject the DNS over TLS server certificate since it would not match their intended server, this could still have the intended result. The client may fall back to traditional DNS queries if DNS over TLS validation fails.

34.6 Redirecting Client DNS Requests

To restrict client DNS to only the DNS Resolver or Forwarder on pfSense® software, use a port forward to capture all DNS requests clients send to other servers.

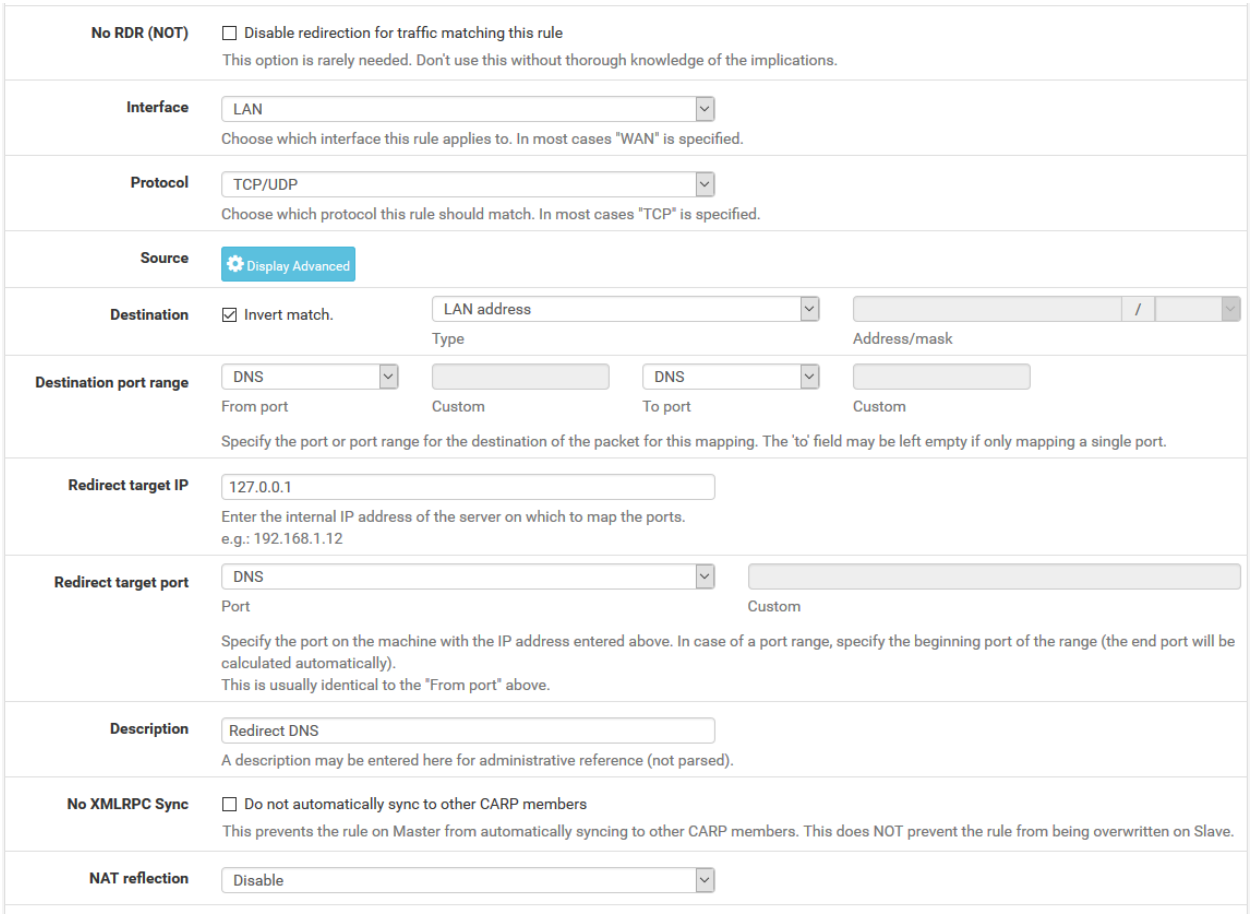
Note: Either The DNS Forwarder or DNS Resolver must be active and it must bind to and answer queries on *Localhost*, or *All* interfaces.

The following example uses the LAN interface, but the same technique will work with any local interface.

- Navigate to **Firewall > NAT**, **Port Forward** tab
- Click  **Add** to create a new rule
- Fill in the following fields on the port forward rule:
  - **Interface**: *LAN*
  - **Protocol**: *TCP/UDP*
  - **Destination**: *Invert Match* **checked**, *LAN Address*
  - **Destination Port Range**: *DNS (53)*
  - **Redirect Target IP**: `127.0.0.1`
  - **Redirect Target Port**: *DNS (53)*
  - **Description**: `Redirect DNS`

- **NAT Reflection**: *Disable*

When complete, the port forward must appear as follows:



Note: If DNS requests to other DNS servers are blocked, such as by following *Blocking External Client DNS Queries*, ensure the rule to pass DNS to `127.0.0.1` is above any rule that blocks DNS.

With this port forward in place, DNS requests from clients made to **any** external IP address will result in the query being answered by the firewall itself. Access to other DNS servers on port 53 is impossible.

Tip: This can be adapted to allow access to only a specific set of DNS servers by changing the Destination network from “LAN Address” to an alias containing the allowed DNS servers. The **Invert match** box should remain checked.

Warning: Clients using DNS over TLS or DNS over HTTPS could circumvent this protection. See *Blocking External Client DNS Queries* for details.

34.7 Dynamic Routing Protocol Basics

Three routing protocols are supported in pfSense® software using the FRR package:

- *BGP* (Border Gateway Protocol)
- *OSPF* (Open Shortest Path First v2, for IPv4).
- *OSPF6* (Open Shortest Path First v3, for IPv6).

An in depth discussion of routing protocols is outside the scope of this documentation.

All of the supported dynamic routing protocols are handled by the *FRR package*. To install FRR:

- Navigate to **System > Package Manager**
- Click **Available Packages**
- Locate **FRR** in the list, or search for it
- Click the  **Install** to the right of the **FRR** package entry.
- Click  **Confirm**
- Wait for the installation to complete
- Navigate to **Services > FRR Global/Zebra**

34.7.1 BGP

The *FRR package documentation* contains a basic *BGP example* which can be used as the basis for most common configurations.

The general form of configuration for FRR BGP is:

- Add a route map to allow routes to be filtered
- Configure the BGP options for the local AS
- Add neighbors

34.7.2 OSPF

The *FRR package documentation* contains a basic *OSPF example* which can be used as the basis for most common configurations.

The general form of configuration for OSPF in FRR is:

- Add interfaces as needed, with local interfaces being marked passive, and those facing other OSPF routers as active.
- Configure the general settings as needed with the router ID, area ID, and so on.

See also:

OpenVPN Site-to-Site with Multi-WAN and OSPF contains an example configuration of OSPF.

34.7.3 OSPF6

The *FRR package documentation* contains a basic *OSPF6 example* which can be used as the basis for most common configurations.

The general form of configuration for OSPF6 in FRR is:

- Add interfaces as needed, with local interfaces being marked passive, and those facing other OSPF6 routers as active.
- Configure the general settings as needed with the router ID, area ID, and so on.

34.8 Basic Firewall Configuration Example

This article is designed to describe how pfSense® software performs rule matching and a basic strict set of rules. The approach described in this document is not the most secure, but will help show how rules are setup.

Rules on the **Interface** tabs are matched on the **incoming** interface.

See also:

Read the *Aliases* article as it will make management of rules easier.

34.8.1 Basic lock down of the LAN and DMZ outgoing rules

Outbound LAN

Make sure the **Default LAN > any** rule is either disabled or removed.

1. Allowing DNS access:
   - If pfSense is the DNS server:
     - Allow **TCP/UDP 53** (DNS) from LAN subnet to **LAN Address**.
   - If using Upstream DNS Servers:
     - Allow **TCP/UDP 53** (DNS) from LAN subnet to **Upstream DNS Servers**.
   - Otherwise:
     - Allow **TCP/UDP 53** (DNS) from LAN subnet to **anywhere**.
2. Allowing all users to browse web pages anywhere:
   - Allow **TCP 80** (HTTP) from LAN subnet to **anywhere**.
3. Allowing users to browse secure web pages anywhere:
   - Allow **TCP 443** (HTTPS) from LAN subnet to **anywhere**.
4. Allowing users to access FTP sites anywhere:
   - Allow **TCP 21** (FTP) from LAN subnet to **anywhere**.
5. Allowing users to access SMTP on a mail server somewhere:
   - Allow **TCP 25** (SMTP) from LAN subnet to **anywhere**.
6. Allowing users to access POP3 on a mail server somewhere:
   - Allow **TCP 110** (POP3) from LAN subnet to **anywhere**.

7. Allowing users to access IMAP on a mail server somewhere:
    - Allow **TCP 143** (IMAP) from LAN subnet to **anywhere**.
8. Allowing remote connections to an outside windows server for remote administration:
    - Allow **TCP/UDP 3389** (Terminal server) from LAN subnet to **IP address of remote server**.
9. Allowing LAN to access windows shares on the DMZ, via NETBIOS/Microsoft-DS:
    - Allow **TCP/UDP 137** from LAN subnet (NETBIOS) to **DMZ subnet**.
    - Allow **TCP/UDP 138** from LAN subnet (NETBIOS) to **DMZ subnet**.
    - Allow **TCP/UDP 139** from LAN subnet (NETBIOS) to **DMZ subnet**.
    - Allow **TCP 445** from LAN subnet (NETBIOS) to **DMZ subnet**.

Outbound DMZ

By default, there are no rules on **OPT** interfaces.

1. Allowing servers to use Windows update or browse the **WAN**:
    - Allow **TCP 80** from DMZ subnet (HTTP) to **anywhere**.
    - Allow **TCP 443** from DMZ subnet (HTTP) to **anywhere**.
2. Allow users to connect to an external DNS server:
    - Allow **TCP/UDP 53** from DMZ subnet (DNS) to **IP address of the upstream DNS server(s)**
3. Allowing servers to use a remote time server:
    - If using an upstream remote time server:
        - Allow **UDP 123** from DMZ subnet (NTP) to **IP address of remote time server**.
    - Otherwise:
        - Allow **UDP 123** from DMZ subnet (NTP) to **any**.

34.8.2 Setup isolating LAN and DMZ, each with unrestricted Internet access

The following setup can be used instead if outbound access is more lenient, but still controlled between local interfaces. This assumes all local networks are privately numbered, and that interfaces have already been configured.

Create an alias, **Firewall > Aliases** from the main menu, called `RFC1918` containing `192.168.0.0/16`, `172.16.0.0/12`, and `10.0.0.0/8`.

LAN Configuration

1. For DNS from the firewall:
    - Allow **TCP/UDP** from LAN subnet to **LAN Address port 53**.
2. For accessing the GUI:
    - Allow **TCP** from LAN subnet to **LAN address port 443**.
3. To ping the firewall from the LAN:
    - Allow **ICMP** from LAN subnet to **LAN address**.

4. If there is any traffic required from LAN to DMZ:
    - Allow any traffic required from **LAN** to **DMZ**.
5. Do not allow LAN to reach DMZ or other private networks:
    - Reject **Any** from LAN subnet to **RFC1918**.
6. For internet access:
    - Allow **Any** from LAN subnet to **any**.

DMZ Configuration

1. For DNS from the firewall:
    - Allow **TCP/UDP** from DMZ subnet to **DMZ Address port 53**.
2. For accessing the GUI (optional):
    - Allow **TCP** from DMZ subnet to **DMZ address port 443**.
3. To ping the firewall from the DMZ:
    - Allow **ICMP** from DMZ subnet to **DMZ address**.
4. If there is any traffic required from DMZ to LAN:
    - Allow any traffic required from **DMZ** to **LAN**.
5. Do not allow DMZ to reach LAN or other private networks:
    - Reject **Any** from DMZ subnet to **RFC1918**.
6. For Internet access:
    - Allow **Any** from DMZ subnet to **any**.

Additional Interfaces

Repeat the above pattern as needed.

34.9 External User Authentication Examples

There are countless ways to configure the user manager to connect to an external RADIUS or LDAP server, but there are some common methods that can be helpful to use as a guide. The following are all tested/working examples, but the server setup will likely vary from the example.

See also:

Authentication Servers

34.9.1 RADIUS Server Example

This example was made against FreeRADIUS but doing the same for Windows Server would be identical. See *Authenticating from Active Directory using RADIUS/NPS* for info on setting up a Windows Server for RADIUS.

This assumes the RADIUS server has already been configured to accept queries from this firewall as a client with a shared secret.

Descriptive Name `ExCoRADIUS`

Type *Radius*

Hostname or IP Address `192.2.0.5`

Shared Secret `secretsecret`

Services Offered *Authentication and Accounting*

Authentication Port `1812`

Accounting Port `1813`

Authentication Timeout `10`

34.9.2 OpenLDAP Example

In this example, the firewall is connecting back to an OpenLDAP server for the company.

Descriptive Name `ExCoLDAP`

Type *LDAP*

Hostname or IP Address `ldap.example.com`

Port `636`

Transport *SSL - Encrypted*

Peer Certificate Authority *ExCo CA*

Protocol Version *3*

Search Scope *Entire Subtree* , `dc=pfsense,dc=org`

Authentication Containers `CN=pfsgroup;ou=people,dc=pfsense,dc=org`

Bind Credentials Anonymous binds *Checked*

Initial Template *OpenLDAP*

User Naming Attribute `cn`

Group Naming Attribute `cn`

Group Member Attribute `memberUid`

RFC2307 Groups Checked

Group Object Class `posixGroup`

UTF8 Encode *Checked*

Username Alterations *Unchecked*

34.9.3 Active Directory LDAP Example

In this example, the firewall connects to an Active Directory structure in order to authenticate users for a VPN. The results are restricted to the **VPNUsers** group. Omit the **Extended Query** to accept any user.

Descriptive Name `ExCoADVPN`

Type *LDAP*

Hostname or IP Address `192.0.2.230`

Port `389`

Transport *TCP - Standard*

Protocol Version *3*

Search Scope *Entire Subtree* , `DC=domain,DC=local`

Authentication Containers `CN=Users,DC=domain,DC=local`

Extended Query memberOf=CN=VPNUsers,CN=Users,DC=example,DC=com

Bind Credentials Anonymous binds *Unchecked*

User DN `CN=binduser,CN=Users,DC=domain,DC=local`

Password `secretsecret`

Initial Template *Microsoft AD*

User Naming Attribute `samAccountName`

Group Naming Attribute `cn`

Group Member Attribute `memberOf`

This example uses plain TCP, but if the Certificate Authority for the AD structure is imported under the Certificate Manager, the connection can also use SSL as well by selecting that option and choosing the appropriate CA from the **Peer Certificate Authority** drop down, and setting the **Hostname** to the match the server certificate.

34.10 Using an External Wireless Access Point

Most SOHO-style wireless routers can be used as an access point if a true Access Point (AP) is not available. If pfSense® software replaced an existing wireless router, the old router can still be used to handle the wireless portion of the network.

This type of deployment is popular for wireless because it is easier to keep the access point in a location with better signal and take advantage of more current wireless hardware without relying on driver support in pfSense software. This way a network supporting 802.11ac or later wireless standards may still be used and secured by pfSense software at the edge, even though pfSense software does not yet have support for newer standards.

This technique is also common with wireless equipment running *WRT, Tomato, or other custom firmware for use as dedicated access points rather than edge routers.

34.10.1 Turning a wireless router into an access point

To convert the wireless router into a wireless access point, follow these generic steps for any device. To find specifics for a particular wireless router, refer to its documentation.

Disable the DHCP server

Disable the DHCP server on the wireless router to prevent a conflict. pfSense software will handle this function for the network, and having two DHCP servers on the same broadcast domain will not function correctly.

Change the LAN IP address

A functional, unique, IP address on the access point is required for management purposes.

Change the LAN IP address on the wireless router to an unused IP address in the subnet where it will reside (commonly LAN). If the firewall running pfSense software replaced this wireless router, then the wireless router was probably using the same IP address now assigned to the firewall LAN interface, which conflicts.

Plug in the LAN interface

Most wireless routers bridge their wireless network to an internal LAN port or switch ports. This means the wireless segment will be on the same broadcast domain and IP subnet as the wired ports. For routers with an integrated switch, any of the LAN switch ports will typically work.

Note: Do not plug in the WAN or Internet port on the wireless router! This will put the wireless network on a different broadcast domain from the rest of the network and the wireless router will perform NAT on the traffic between the wireless and LAN. This also results in double NAT of traffic between the wireless network and the Internet. This is an ugly design, and will lead to problems in some circumstances, especially if communication must occur between wireless and wired LAN clients.

Deciding where to connect the LAN interface from the wireless router depends on the chosen network design. The next sections cover options and considerations for selecting the best deployment style.

34.10.2 Bridging wireless to the LAN

One common means of deploying wireless is to plug the access point directly into the same switch as the LAN hosts, where the AP bridges the wireless clients onto the wired network. This works well, but offers limited control over the ability of the wireless clients to communicate with internal hosts.

See also:

See *Choosing Routing or Bridging* for details on bridging in this role.

34.10.3 Bridging wireless to an OPT interface

To keep wireless and wired networks on the same IP subnet and broadcast domain while also increasing control over wireless clients, add an OPT interface to the firewall for the access point and bridge the OPT interface to the LAN interface.

Warning: Though bridging offers increased control over traffic, it also results in lower performance as all wireless traffic must pass through and be processed by the firewall. Typically, wireless speeds are low enough that this is not a major concern, but as wireless speeds improve the severity of the problem also increases.

This scenario is functionally equivalent to plugging the access point directly into the LAN switch, except pfSense software can filter traffic from the wireless network to provide protection to LAN hosts and vice versa.

Note: A configuration with the bridge assigned as LAN is optimal here, rather than only having the OPT bridged to the existing wired LAN.

34.10.4 Routed segment on an OPT interface

The wireless network can also be placed on a separate IP subnet if desired. This is done without bridging the OPT interface on pfSense, instead assigning it with an IP address in a separate subnet different from the LAN. This enables routing between internal and wireless networks, as permitted by the firewall ruleset. This is commonly done on larger networks, where multiple access points are plugged into a switch that is then plugged into the OPT interface on pfSense. It is also preferable when wireless clients will be forced to connect to a VPN before allowing connections to internal network resources.

34.11 Using Software from FreeBSD

pfSense® software is based on FreeBSD, thus many familiar FreeBSD packages are available for use by veteran FreeBSD system administrators.

Warning: Installing software this way **will** have unintended side effects. This action is not recommended or supported by Netgate.

Many parts of FreeBSD are not included in the base installation of pfSense software, so library and other issues can occur when attempting to use software installed in this manner. The pfSense software base installation does not include a compiler in the base system for many reasons, and as such software cannot be built locally. However, packages can be installed from FreeBSD the package repository.

34.11.1 Concerns/Warnings

Several important concerns must be considered by any administrator before deciding to install additional software, especially software that is not obtained from Netgate package repositories.

Security Concerns

Any extra software added to a firewall is a security problem, and must be evaluated fully before installation. If the need outweighs the risk, it may be worth taking. Official packages for pfSense software are not immune to this problem either. Any additional service is another potential attack vector.

Performance Concerns

Most hardware running pfSense software can handle the traffic load with which it is tasked. If the firewall hardware has horsepower to spare, it may not hurt performance to add additional software. That said, be mindful of the resources consumed by the added software.

Conflicting Software

If an installed package duplicates functionality found in the base system, or replaces a base system package with a newer version, it could cause unpredictable system instability. Ensure that the software does not already exist in pfSense.

Lack of Integration

Any extra software installed will not have GUI integration. For some, this is not a problem, but there have been people who expected to install a package and have a GUI magically appear for its configuration. These packages must be configured by hand. If this is a service, that means also making sure that any startup scripts accommodate the methods used by pfSense software.

Software can also install additional web pages that are not protected by the pfSense software authentication process. Test any installed software to ensure that it protects and filters access appropriately.

Lack of Backups

Packages installed in this manner must have any configuration or other needed files backed up manually.

These files will not be backed up during a normal backup and could be lost or changed during a firmware update. The add-on package described in *Backup Files and Directories with the Backup Package* is capable of backing up arbitrary files such as these.

34.11.2 Installing Packages

To install a package, the proper package site must be used. pfSense software is compiled against a specific FreeBSD branch, and has only a specific set of packages hosted on Netgate servers.

Packages located in the Netgate package repository, including some FreeBSD software packages that are not a part of the pfSense software distribution, can be installed using `pkg install` directly:

```
# pkg install screen
```

Or use a full URL to a `pkg add` to add them from the FreeBSD package servers:

```
# pkg add http://pkg.freebsd.org/FreeBSD:11:amd64/latest/All/tshark-3.2.6.txz
```

The `pkg` utility will download and install the package, along with its required dependencies.

Additionally, the full set of FreeBSD packages can be made available by editing `/usr/local/etc/pkg/repos/pfSense.conf` and changing the first line to:

```
FreeBSD: { enabled: yes }
```

Warning: Adding software from FreeBSD package repositories **will** introduce problems with package dependencies, especially if a package depends on another piece of software that already exists on the firewall which may have been built with conflicting options. Take extreme caution when adding packages in this way.

Custom packages can also be built on another computer running FreeBSD and then the package file can be copied and installed on a firewall running pfSense software. Due to the complexity of this topic, it will not be covered here.

34.11.3 Maintaining Packages

The following command prints a list of all currently installed packages, including packages and components of the base system of pfSense software:

```
# pkg info
```

To delete an installed package, pass its full name or use a wildcard:

```
# pkg_delete iftop-1.0.p4
# pkg_delete pstree-\*
```

34.12 Using EAP and PEAP with FreeRADIUS

34.12.1 General EAP configuration

The default EAP settings will work in most situations (EAP-MD5, EAP-TLS, EAP-TTLS, EAP-PEAP) so there is no need to change them without any need. If EAP-TTLS or EAP-PEAP is used with VLAN assignment then set **Use Tunneled Reply** to *yes*:

To make the use of certificates more secure, check the **Common Name** of the client certificate against the username entered in **FreeRADIUS > Users**. For this set **Check Client Certificate CN** to *yes*:

Another option to increase security with certificates is to check the issuer of the client certificate against the CA certificate. This can be enabled with **Check Cert Issuer** but then it is necessary to enter country, state, province and organization - case sensitive - to match the CA.

FreeRADIUS by default allows many EAP types for authentication. In some environments only some strong EAP types (TLS, TTLS, PEAP, MSCHAPv2) may be allowed or weak types (MD5, GTC, LEAP) may be disallowed. Disable the weak EAP types in FreeRADIUS using **Disable weak EAP types** so that FreeRADIUS rejects users which try to authenticate using such a weak method. If these types are disabled it does **not** affect the inner tunnel session in EAP-TTLS and EAP-PEAP. Further it is no problem to use a weak or cleatext method in the inner tunnel because if the outer tunnel uses one of the above call strong encryption types.

FreeRADIUS is multitalented. It can handle almost all authentication types hosts send. So if weak encryption types such as MD5 and others are not disabled then the following will happen:

```
a client wants to authenticate using MD5 => freeradius will do that
a client wants to authenticate using LEAP => freeradius will do that
a client wants to authenticate with TLS or TTLS or PEAP or MSCHAPv2 or ... =>␣
↪freeradius will do that
```

So **disable weak encryption** is checked then this disables MD5, GTC and LEAP. This will happen:

```
a client wants to authenticate using MD5 -> freeradius will not do that because that␣
↪was disabled
a client wants to authenticate using LEAP -> freeradius will not do that because that␣
↪was disabled
a client wants to authenticate with TLS or TTLS or PEAP or MSCHAPv2 or... ->␣
↪freeradius will do that.
```

34.12.2 PEAP and MSCHAPv2

- FreeRADIUS package configuration:
  - Configure an interface in **FreeRADIUS > Interfaces**
  - Create a **CA-Certificate** and a **Server-Certificate**. Choose pfSense® Cert-Manager or FreeRADIUS Cert-Manager but **never** use the default certificates which come with FreeRADIUS after package installation!
  - Select the certificates in **FreeRADIUS > EAP**. If FreeRADIUS as Cert-Manager is selected then nothing needs changed. If pfSense Cert-Manager was chosen, then it must be enabled there and the certs must be chosen from the pulldown menu. Click **Save**.
  - Add the WLAN-AccessPoint in **FreeRADIUS > NAS/Clients**
  - Add a username/password in **FreeRADIUS > Users**
- WLAN Access-Point Configuration:
  - Change the wireless encryption to **WPA-Enterprise** or better **WPA2-Enterprise** with **TKIP** or better **AES/CCMP**
  - Do **not** use a passphrase but select **RADIUS** or **802.1X**. Enter the IP-Address of the pfSense FreeRADIUS-Server and the shared secret according to that what was entered in **FreeRADIUS > NAS/Clients**
- WLAN Device (Supplicant) Configuration:
  - Some devices can autoconfigure the Authentication- and Encryption-Method. If not choose **PEAP as encryption** and **MS-CHAPv2 as Authentication**. 1
  - Connect to WLAN AccessPoint and the client will be prompted for username and password
  - Some devices auto-accept the CA-Certificate as valid. Often this CA-Certificate will first need to be accepted. This is the certificate created on pfSense.

The most part of the "command line action" which is done in these tutorials can be done from FreeRADIUS GUI.

- A very good how-to with screenshots in german language can be found here.

34.12.3 EAP-TLS

- pfSense configuration:
  - Create a **CA**, a **Server-Certificate** and a **Client-Certificate**. Using **System > Cert Manager** is recommended.
- FreeRADIUS configuration:
  - Create an **interface**, add a **NAS/Client** and create a **user**. For this example, use *myuser* as username and *mypass* as password.
  - The EAP default options are working - read *FreeRADIUS package*.
  - Using **pfSense Cert-Manager** and selecting the CA and the server certificate is recommended.
  - Leave the password field empty
  - Download the CA.crt - **not the key** - from **System > Cert Manager**, **CAs** tab and Client *.p12* from **System > Cert Manager**, **Certificates** tab
- Client Requires password on .p12
  - If a client will not load the .p12 without a password on it, and space does not work, add a password with openssl
  - Download user cert and key vs the p12 and with the ca cert use the following command
  - `openssl pkcs12 -export -certfile ca.crt -in user.crt -inkey user.key -out user.p12`

34.13 Using Mobile One-Time Passwords with FreeRADIUS

Using Mobile-One-Time-Password (mOTP) with the *FreeRADIUS package*.

34.13.1 Enable Mobile-One-Time-Password (OTP) support

This documentation will cover many parts from installation, configuration, modification, and more from here.

A one time password is a password which can be only used one time and will be only usable within a short time period (10s). So it can be compared with the handling of tokens from RSA SecureID.

This kind of password generation makes sense in some scenarios but not in all. It probably makes no sense to use these passwords in the office - there shouldn't be any attacker. But the mOTP could make sense for Road Warriors. Most of them use an state of the art mobile phone and a notebook to connect to the company VPN. For more take a look at the chapter below: *Miscellaneous configuration and hints*

pfSense® software configuration:

- Enable Mobile-One-Time-Password in **FreeRADIUS > Settings**

This will install the script in */usr/local/bin/otpverify.sh*. To execute the script we need the additional package *bash*. This will be installed automatically.

FreeRADIUS configuration:

- Create a user in **FreeRADIUS > Users**
- Enter a **username** but **do not** enter any password!
- Check **Enable Mobile-One-Time-Password For This User**

- Enter the **Init-Secret**. The init-secret will be created on the client (mobile device, mobile phone)
- Enter the **PIN**. Every time the user wants to generate a new password with his mobile token then he has to enter the PIN and then the token generates a one-time-password.
- The **Offset** should be zero by default. If the mobile token is on another time zone than the FreeRADIUS server then correct the offset. If the mobile device changes its time zone automatically than there is no need to do this.

Miscellaneous configuration and hints:

- The configuration for different mobile devices can be found here.
- And here is the software for mobile devices.
- Please read the limitations to make sure how secure this is.
- mOTP will probably not work with EAP, CHAP, MSCHAP. If it does - tell me how :-)
- A generated one-time-password can be used for ~20 seconds. For example the Windows token generator generates new tokens every ~10 seconds. Perhaps other token generators use other timespans for generating tokens. FreeRADIUS and the OTP script accept tokens which were generated within the last 20 seconds.

34.14 Using NAT and FTP without a Proxy

pfSense® firewalls do not include an FTP Proxy, this doesn't affect clients and servers as much as one might think.

Important: Use of FTP is strongly discouraged. It is an outdated protocol that transmits credentials and other data openly without encryption which is very insecure.

34.14.1 Client Behind pfSense

FTPS, or encrypted FTP, is not affected. The proxy could not have affected its traffic before.

A client on a LAN or other internal interface behind a pfSense firewall will likely not notice any difference. Most clients, aside from the Microsoft command line FTP program, default to passive (PASV) FTP, where clients make outbound connections to servers.

Passive mode on the client will require access to random/high ports outbound, which could run afoul of a strict outbound ruleset. Environments with a security policy that requires strict outbound firewall rules likely would not be using FTP anyhow, as it transmits credentials without encryption.

Active mode FTP through NAT will not function as that relies on a proxy or similar mechanism. Use Passive mode instead. Another option is the recently added FTP Client Proxy package which leverages in FreeBSD to allow clients on local interfaces to reach remote FTP servers with active FTP.

Active mode FTP for a client that does not involve NAT (Client has a public IP address) should work so long as WAN rules pass the appropriate traffic back to the client. The client may have a configurable active port range to make that simpler.

34.14.2 Server Behind pfSense

FTPS, or encrypted FTP, is not affected. The proxy could not have affected its traffic before.

A server behind pfSense would work fine with active mode, there would be no difference here. In active mode the server would make outbound connections back to the client, so as long as the firewall rules on the interface containing the server allow outbound connections, it will work.

A server behind pfSense running in Passive mode will function but requires a few items to be configured:

1. Port forwards or 1:1 NAT to forward not only port 21, but also the passive port range in to the server
2. The passive port range must be configured on the server, corresponding to the range of ports forwarded in the previous step.
3. The server may also need to be configured to account for NAT. Some clients will ignore private addresses in passive responses so this may not be necessary.

Sample Configuration for vsftpd

In **vsftpd.conf**:

```
# Do not allow the client to use PORT
port_enable=NO
# Use the hostname in the PASV response (DNS must be setup and match!)
pasv_addr_resolve=YES
# Enable Passive Mode
pasv_enable=YES
# Set the passive port range (1000 ports)
pasv_min_port=20000
pasv_max_port=20999
```

34.15 Configuring pfSense Software for Online Gaming

This page provides information on using pfSense® software with online games.

First, many games will require the use of *Static Port* or *UPnP/NAT-PMP*.

The forum often has a wide array of threads for specific games and consoles. Search there for games if they are not listed here.

34.15.1 Specific Game/Console Information

Recommendations for specific games can be found below. If any special handling is required but not listed here, please *submit a documentation update*. Include a link to manufacturer's documentation when possible.

NB: What works to make a single console/device work from behind a firewall may not work for multiple consoles/devices behind the same firewall.

Nintendo Wii/Wii U/3DS

No special configuration is typically required, though some cases may need UPnP to work. See *What are UPnP and NAT-PMP*.

Nintendo Switch

Static Port should be setup. See *Static Port*.

Steam

Varies by game, but typically *UPnP/NAT-PMP* or manual port forwards are enough to make it work. Some may require *Static Port*.

Gunz Online

To use multiple machines behind pfSense to play, configure each Gunz Online machine with a different port. Visit **NAT > Outbound** and setup a custom port entry for each machine using the appropriate custom port.

Xbox One

Static Port should be setup. UPnP may also be required in some cases. See *Static Port*, *UPnP/NAT-PMP*. Xbox Support

34.16 High Availability Configuration Example

This recipe describes a simple three interface HA configuration. The three interfaces are LAN, WAN, and Sync. This is functionally equivalent to a two interface LAN and WAN deployment, with the Sync interface being used to synchronize configuration and firewall states between the primary and secondary firewalls.

Note: This example only covers an IPv4 configuration. High Availability is compatible with IPv6, but it requires static addressing on the firewall interfaces. When preparing to configure HA, if static IPv6 assignments are not available, set IPv6 to *None* on all interfaces.

See also:

Review *High Availability* before following this recipe.

For help diagnosing issues with HA, see:

- *Troubleshooting High Availability*
- *Troubleshooting High Availability Clusters in Virtual Environments*
- *Troubleshooting VPN Connectivity to a High Availability Secondary Node*

34.16.1 Determine IP Address Assignments

The first task is to plan IP address assignments. A good strategy is to use the lowest usable IP address in the subnet as the CARP VIP, the next subsequent IP address as the primary firewall interface IP address, and the next IP address as the secondary firewall interface IP address. This design is optional, any scheme may be used, but we strongly recommend a consistent and logical scheme to make design and administration simpler.

WAN Addressing

The WAN addresses will be selected from those assigned by the ISP. For the example in Table *WAN IP Address Assignments*, the WAN of the HA pair is 198.51.100.0/24, and the addresses 198.51.100.200 through 198.51.100.202 will be used as the WAN IP addresses.

Table 1: WAN IP Address Assignments

IP Address	Usage
198.51.100.200/24	CARP shared IP address
198.51.100.201/24	Primary node WAN IP address
198.51.100.202/24	Secondary node WAN IP address

LAN Addressing

The LAN subnet is 192.168.1.0/24. For this example, the LAN IP addresses will be assigned as shown in Table *LAN IP Address Assignments*.

Table 2: LAN IP Address Assignments

IP Address	Usage
192.168.1.1/24	CARP shared IP address
192.168.1.2/24	Primary node LAN IP address
192.168.1.3/24	Secondary node LAN IP address

Sync Interface Addressing

There is no shared CARP VIP on this interface because there is no need for one. These IP addresses are used only for communication between the firewalls. For this example, 172.16.1.0/24 is used as the Sync subnet. Only two IP addresses will be used, but a /24 is used to be consistent with the other internal interface (LAN). For the last octet of the IP addresses, use the same last octet as that firewall's LAN IP address for consistency.

Table 3: Sync IP Address Assignments

IP Address	Usage
172.16.1.2/24	Primary node Sync IP address
172.16.1.3/24	Secondary node Sync IP address

Figure *Example HA Network Diagram* shows the layout of this example HA pair. The primary and secondary each have identical connections to the WAN and LAN, and a crossover cable between them to connect the Sync interfaces. In this basic example, the WAN switch and LAN switch are still potential single points of failure. Switching redundancy is covered later in this chapter in *Layer 2 Redundancy*.

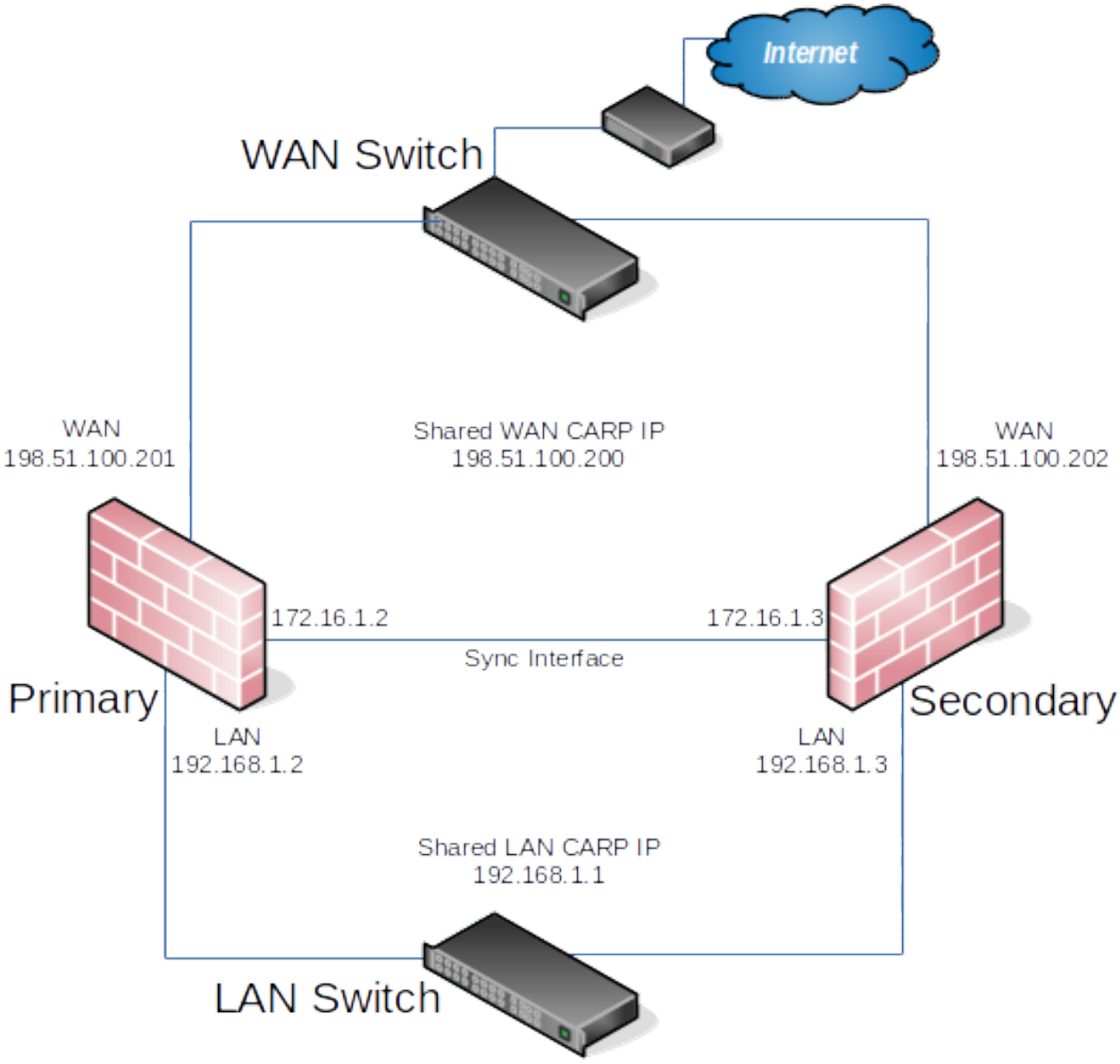


Fig. 5: Example HA Network Diagram

34.16.2 Cluster Configuration Basics

Each node requires some basic configuration outside of the actual HA setup. Do not connect both nodes into the same LAN before both nodes have a non-conflicting LAN setup.

Installation, interface assignment and basic configuration

Install the OS on the firewalls as usual and assign the interfaces identically on both nodes. Interfaces must be assigned in the same order on all nodes exactly. If the interfaces are not aligned, configuration synchronization and other tasks will not behave correctly. If any adjustments have been made to the interface assignments, they must be replicated identically on both nodes.

Then, connect to the GUI and use the Setup Wizard to configure each firewall with a unique hostname and non-conflicting static IP addresses. Refer back to *Setup Wizard* if needed.

For example, one node could be "firewall-a.example.com" and the other "firewall-b.example.com", or a more personalized pair of names.

Note: Avoid naming the nodes "master" or "backup" since those are states and not roles, instead they could be named "primary" and "secondary".

The default LAN IP address is 192.168.1.1. Each node must be moved to its own address, such as 192.168.1.2 for the primary and 192.168.1.3 for the secondary. This layout is shown in *LAN IP Address Assignments*. Once each node has a unique LAN IP address, then both nodes may be plugged into the same LAN switch.

34.16.3 Setup Sync Interface

Before proceeding, the Sync interfaces on the cluster nodes must be configured. *Sync IP Address Assignments* lists the addresses to use for the Sync interfaces on each node. Once that has been completed on the primary node, perform it again on the secondary node with the appropriate **IPv4 address** value.

To complete the Sync interface configuration, firewall rules must be added to both nodes to allow synchronization.

At a minimum, the firewall rules must pass the configuration synchronization traffic (by default, HTTPS on port 443) and pfsync traffic. In most cases, a simple "allow all" style rule is used.

When complete, the rules will look like the example in figure *Example Sync Interface Firewall Rules*, which also includes a rule to allow ICMP echo (ping) for diagnostic purposes.

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐ ✔	0/0 B	IPv4 TCP	SYNC net	*	SYNC address	443 (HTTPS)	*	none		Allow configuration synchronization
☐ ✔	0/11 KiB	IPv4 PFSYNC	SYNC net	*	*	*	*	none		Allow state synchronization
☐ ✔	0/0 B	IPv4 ICMP echoreq	SYNC net	*	SYNC net	*	*	none		Allow ICMP echo (ping) for Diagnostics

Fig. 6: Example Sync Interface Firewall Rules

The secondary does not need those rules initially, only a rule to allow traffic to the GUI for XML-RPC to function. The full set of rules will synchronize once XML-RPC has been configured.

34.16.4 Configure pfsync

State synchronization using pfsync must be configured on both the primary and secondary nodes to function.

First on the primary node and then on the secondary, perform the following:

- Navigate to **System > High Avail Sync**
- Check **Synchronize States**
- Set **Synchronize Interface** to *SYNC*
- Set **pfsync Synchronize Peer IP** to the other node. Set this to `172.16.1.3` when configuring the primary node, or `172.16.1.2` when configuring the secondary node
- Click **Save**

34.16.5 Configure Configuration Synchronization (XML-RPC)

Warning: Configuration synchronization **must only be configured on the primary node**. Never activate options in this section on the secondary node of a two-member cluster.

On the primary node only, perform the following:

- Navigate to **System > High Avail Sync**
- Set **Synchronize Config to IP** to the Sync interface IP address on the secondary node, `172.16.1.3`
- Set **Remote System Username** to `admin`.

  Note: This must be `admin`, or the same user on both nodes with the "System - HA node sync" privilege

- Set **Remote System Password** to the admin user account password, and repeat the value in the confirmation box.
- Check the boxes for each area to synchronize to the secondary node. For this guide, as with most configurations, all boxes are checked. The **Toggle All** button may be used to select all of the options at once, rather than selecting them individually.
- Click **Save**

As a quick confirmation that the synchronization worked, on the secondary node navigate to **Firewall > Rules** on the **SYNC** tab. The rules entered on the primary are now there, and the temporary rule is gone.

The two nodes are now linked for configuration synchronization! Changes made to the primary node in supported areas will be synchronized to the secondary whenever a change is made.

Warning: Do not make changes to the secondary in areas set to be synchronized! These changes will be overwritten the next time the primary node performs a synchronization.

Configuring the CARP Virtual IPs

With configuration synchronization in place, the CARP Virtual IP addresses need only be added to the primary node and they will be automatically copied to the secondary.

- Navigate to **Firewall > Virtual IPs** on the primary node to manage CARP VIPs
- Click  **Add** at the top of the list to create a new VIP.

Note: A VIP must be added for each interface handling user traffic, in this case WAN and LAN.

Type Defines the type of VIP, in this case *CARP*.

Interface Defines the interface upon which the VIP will reside, such as *WAN*

Address(es) The **Address** box is where the IP address values are entered for the VIP. A subnet mask must also be selected and it must match the subnet mask on the interface IP address. For this example, enter `198.51.100.200` and `24` (See *WAN IP Address Assignments*).

Virtual IP Password Sets the password for the CARP VIP. This need only match between the two nodes, which will be handled by synchronization. The password and confirm password box must both be filled in and they must match.

VHID Group Defines the ID for the CARP VIP A common tactic is to make the VHID match the last octet of the IP address, so in this case choose `200`

Advertising Frequency determines how often CARP heartbeats are sent.

Base Controls how many whole seconds elapse between Heartbeats, typically *1*. This should match between cluster nodes.

Skew Controls fractions of a second (1/256th increments). A primary node is typically set to 0 or 1, secondary nodes will be 100 or higher. This adjustment is handled automatically by XML-RPC synchronization.

Description Some text to identify the VIP, such as `WAN CARP VIP`.

Note: If CARP appears to be too sensitive to latency on a given network, adjusting the **Base** by adding one second at a time is recommended until stability is achieved.

The above description used the WAN VIP as an example. The LAN VIP would be configured similarly except it will be on the *LAN* interface and the address will be `192.168.1.1` (See *LAN IP Address Assignments*).

If there are any additional IP addresses in the WAN subnet that will be used for purposes such as 1:1 NAT, port forwards, VPNs, etc, they may be added now as well.

Click **Apply Changes** after making any edits to the VIPs.

After adding VIPs, check **Firewall > Virtual IPs** on the secondary node to ensure that the VIPs synchronized as expected.

The Virtual IP addresses on both nodes will look like *CARP Virtual IP Address List* if the process was successful.

Virtual IP Address

Virtual IP address	Interface	Type	Description
198.51.100.200/24 (vhid: 200)	WAN	CARP	WAN CARP VIP
192.168.1.1/24 (vhid: 1)	LAN	CARP	LAN CARP VIP

Fig. 7: CARP Virtual IP Address List

Configure Outbound NAT for CARP

The next step will be to configure NAT so that clients on the LAN will use the shared WAN IP as the address.

- Navigate to **Firewall > NAT**, **Outbound** tab
- Click to select *Manual Outbound NAT rule generation*
- Click **Save**

A set of rules will appear that are the equivalent rules to those in place for Automatic Outbound NAT. Adjust the rules for internal subnet sources to work with the CARP IP address instead.

- Click  to the right of the rule to edit
- Locate the **Translation** section of the page
- Select the WAN CARP VIP address from the **Address** drop-down
- Change the Description to mention that this rule will NAT LAN to the WAN CARP VIP address

> **Warning:** If additional local interfaces are added later, such as a second LAN, DMZ, etc, and that interface uses private IP addresses, then additional manual outbound NAT rules must be added at that time.

When complete, the rule changes will look like those found in *Outbound NAT Rules for LAN with CARP VIP*

Mappings

	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description
☐ ✔	WAN	192.168.1.0/24	*	*	500	198.51.100.200	*	✔	Auto created rule for ISAKMP - LAN to WAN
☐ ✔	WAN	192.168.1.0/24	*	*	*	198.51.100.200	*	⇄	Auto created rule - LAN to WAN

Fig. 8: Outbound NAT Rules for LAN with CARP VIP

Modifying the DHCP Server

The DHCP server daemons on the cluster nodes need adjustments so that they can work together. The changes will synchronize from the primary to the secondary, so as with the VIPs and Outbound NAT, these changes need only be made on the primary node.

- Navigate to **Services > DHCP Server**, **LAN*** tab.
- Set the **DNS Server** to the LAN CARP VIP, here `192.168.1.1`
- Set the **Gateway** to the LAN CARP VIP, here `192.168.1.1`
- Set the **Failover Peer IP** to the actual LAN IP address of the secondary node, here `192.168.1.3`
- Click **Save**

Setting the **DNS Server** and **Gateway** to a CARP VIP ensures that the local clients are talking to the failover address and not directly to either node. This way if the primary fails, the local clients will continue talking to the secondary node.

The **Failover Peer IP** allows the daemon to communicate with the peer directly in this subnet to exchange data such as lease information. When the settings synchronize to the secondary, this value is adjusted automatically so the secondary points back to the primary.

34.17 High Availability Configuration Example with Multi-WAN

HA can also be deployed for firewall redundancy in a multi-WAN configuration. This section details the VIP and NAT configuration needed for a dual WAN HA deployment. This section only covers topics specific to HA and multi-WAN.

34.17.1 Determine IP Address Assignments

For this example, four IP addresses will be used on each WAN. Each firewall needs an IP address, plus one CARP VIP for Outbound NAT, plus an additional CARP VIP for a 1:1 NAT entry that will be used for an internal mail server in the DMZ segment.

WAN and WAN2 IP Addressing

Table *WAN IP Addressing* show the IP addressing for both WANs. In most environments these will be public IP addresses.

Table 4: WAN IP Addressing

IP Address	Usage
198.51.100.200	Shared CARP VIP for Outbound NAT
198.51.100.201	Primary firewall WAN
198.51.100.202	Secondary firewall WAN
198.51.100.203	Shared CARP VIP for 1:1 NAT

Table 5: WAN2 IP Addressing

IP Address	Usage
203.0.113.10	Shared CARP VIP for Outbound NAT
203.0.113.11	Primary firewall WAN2
203.0.113.12	Secondary firewall WAN2
203.0.113.13	Shared CARP VIP for 1:1 NAT

LAN Addressing

The LAN subnet is 192.168.1.0/24. For this example, the LAN IP addresses will be assigned as follows.

Table 6: LAN IP Address Assignments

IP Address	Usage
192.168.1.1	CARP shared LAN VIP
192.168.1.2	Primary firewall LAN
192.168.1.3	Secondary firewall LAN

DMZ Addressing

The DMZ subnet is 192.168.2.0/24. For this example, the DMZ IP addresses will be assigned as follows in Table *DMZ IP Address Assignments*.

Table 7: DMZ IP Address Assignments

IP Address	Usage
192.168.2.1	CARP shared DMZ VIP
192.168.2.2	Primary firewall DMZ
192.168.2.3	Secondary firewall DMZ

pfsync Addressing

There will be no shared CARP VIP on this interface because there is no need for one. These IP addresses are used only for communication between the firewalls. For this example, 172.16.1.0/24 will be used as the Sync subnet. Only two IP addresses will be used, but a /24 is used to be consistent with the other internal interfaces. For the last octet of the IP addresses, the same last octet as that firewall's LAN IP is chosen for consistency.

Table 8: Sync IP Address Assignments

IP Address	Usage
172.16.1.2	Primary firewall Sync
172.16.1.3	Secondary firewall Sync

34.17.2 NAT Configuration

The NAT configuration when using HA with Multi-WAN is the same as HA with a single WAN. Ensure that only CARP VIPs are used for inbound traffic or routing. See *Network Address Translation* for more information on NAT configuration.

34.17.3 Firewall Configuration

With Multi-WAN a firewall rule must be in place to pass traffic to local networks using the default gateway. Otherwise, when traffic attempts to reach the CARP address or from LAN to DMZ it will instead go out a WAN connection.

A rule must be added at the top of the firewall rules for all internal interfaces which will direct traffic for all local networks to the default gateway. The important part is the gateway needs to be *default* for this rule and not one of the failover or load balance gateway groups. The destination for this rule would be the local LAN network, or an alias containing any locally reachable networks.

34.17.4 Multi-WAN HA with DMZ Diagram

Due to the additional WAN and DMZ elements, a diagram of this layout is much more complex as can be seen in Figure *Diagram of Multi-WAN HA with DMZ*.

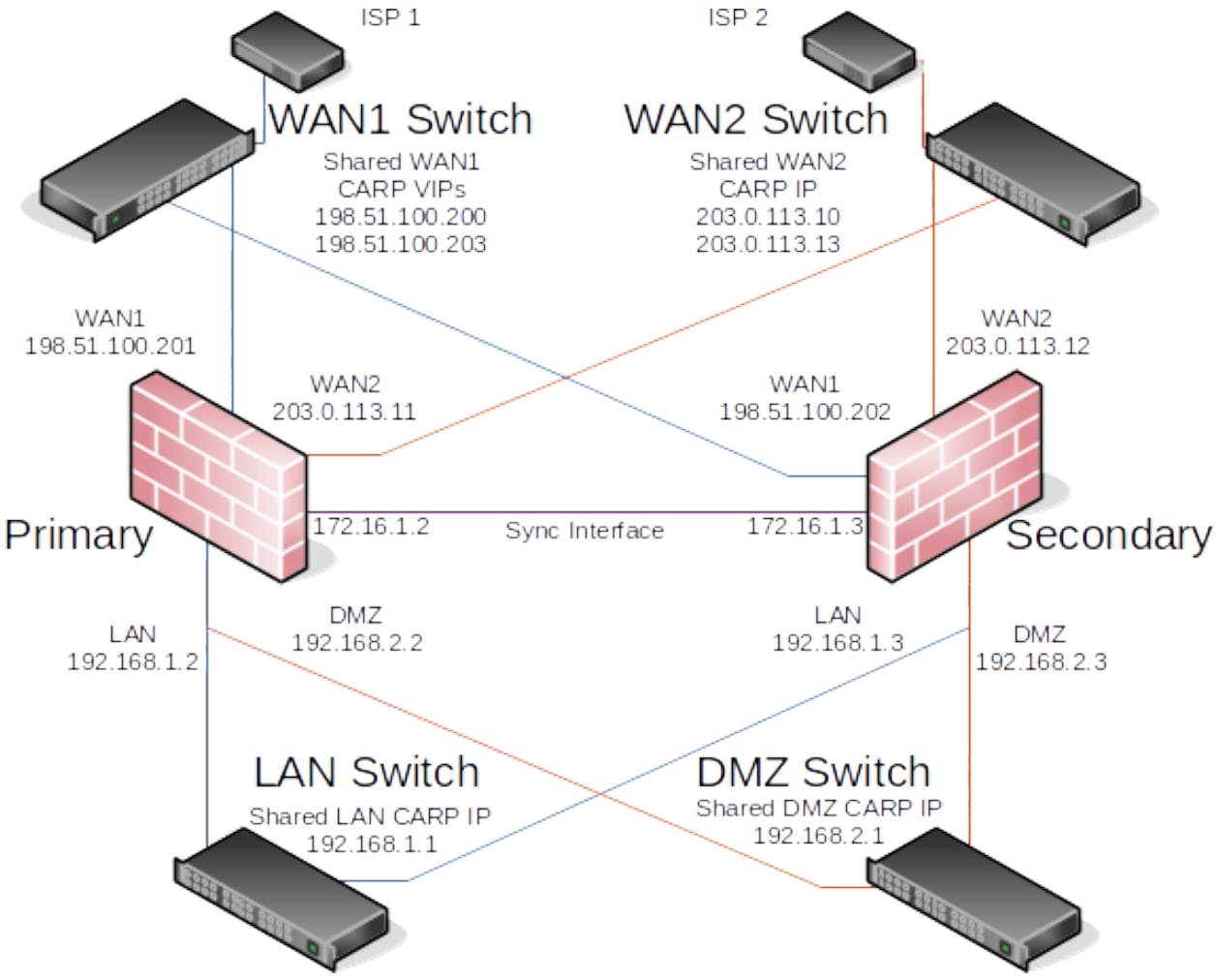


Fig. 9: Diagram of Multi-WAN HA with DMZ

34.18 High Availability Configuration Example without NAT

As mentioned earlier, only CARP VIPs provide redundancy for addresses directly handled by the firewall, and they can only be used in conjunction with NAT or services on the firewall itself. Redundancy can also be provided for routed public IP subnets with HA. This section describes this type of configuration, which is common in large networks, ISP and wireless ISP networks, and data center environments.

34.18.1 Public IP Assignments

At least a /29 public IP block for the WAN side of pfSense® is necessary, which provides six usable IP addresses. Only three are required for a two firewall deployment, but this is the smallest IP subnet that will accommodate three IP addresses. Each firewall requires one IP, and at least one CARP VIP is needed on the WAN side.

The second public IP subnet will be routed to one of the CARP VIPs by the ISP, data center, or upstream router. Because this subnet is being routed to a CARP VIP, the routing will not be dependent upon a single firewall. For the depicted example configuration in this chapter, a /24 public IP subnet will be used and it will be split into two /25 subnets.

34.18.2 Network Overview

The example network depicted here is a data center environment consisting of two pfSense firewalls with four interfaces each: WAN, LAN, DBDMZ, and pfsync. This network contains a number of web and database servers. It is not based on any real network, but there are countless production deployments similar to this.

WAN Network

The WAN side connects to the upstream network, either the ISP, data center, or upstream router.

WEB Network

The WEB segment in this network uses the "LAN" interface but renamed. It contains web servers, so it has been named WEB but it could be called DMZ, SERVERS, or anything desired.

DBDMZ Network

This segment is an OPT interface and contains the database servers. It is common to segregate the web and database servers into two networks in hosting environments. The database servers typically do not require direct access from the Internet, and hence are less subject to compromise than web servers.

Sync Network

The Sync network in this diagram is used to replicate pfSense configuration changes via XML-RPC and for pfsync to replicate state table changes between the two firewalls. As described earlier in this chapter, a dedicated interface for this purpose is recommended.

Network Layout

Figure *Diagram of HA with Routed IPs* illustrates this network layout, including all routable IP addresses, the WEB network, and the Database DMZ.

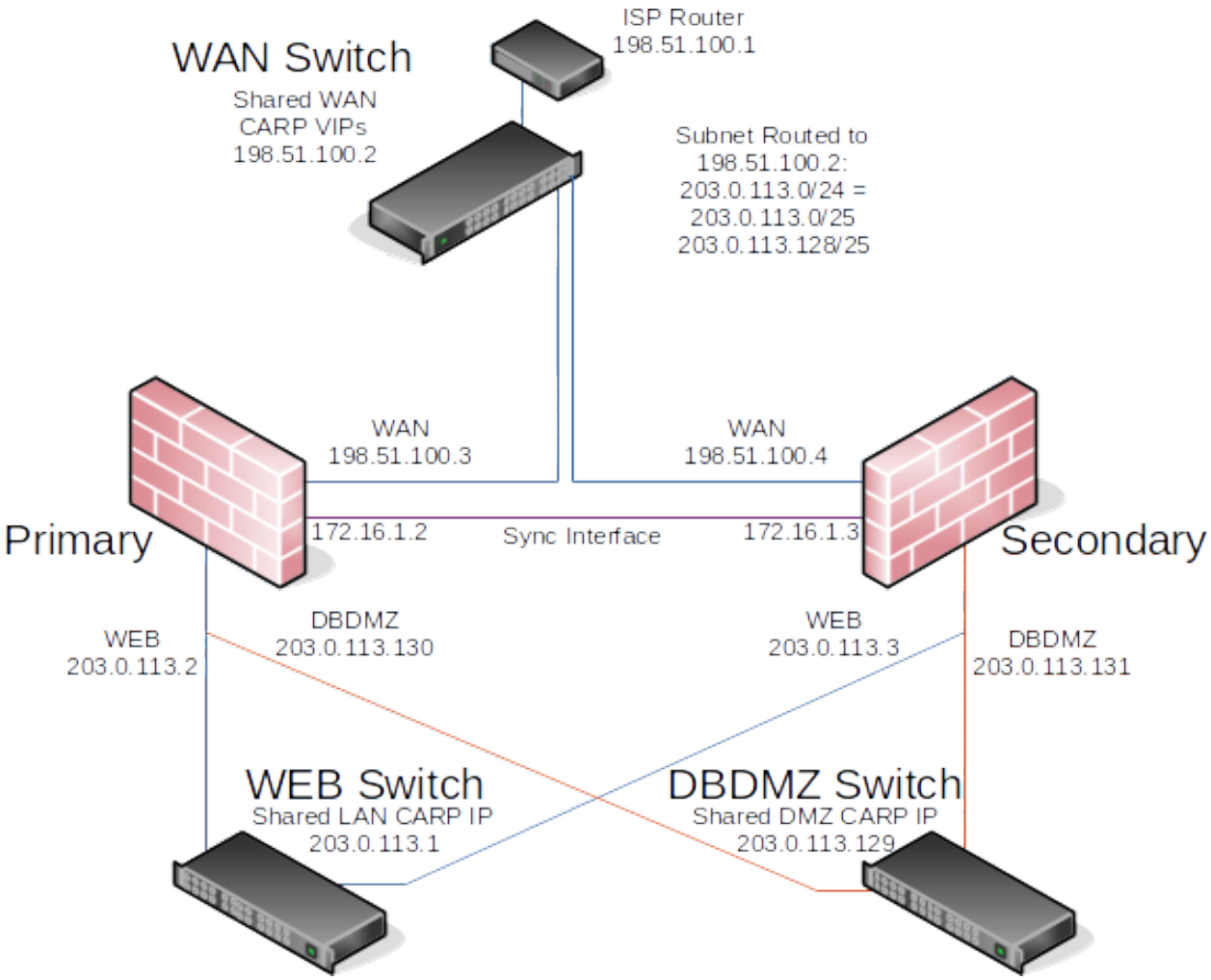


Fig. 10: Diagram of HA with Routed IPs

Note: Segments containing database servers typically do not need to be publicly accessible, and hence would more commonly use private IP subnets, but the example illustrated here can be used regardless of the function of the two internal subnets.

34.19 A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid

This section is not meant to be a formal, detailed, and comprehensive recipe for using Squid and other related web proxy software, but a quick run-through to get it up and running and to cover the most commonly asked questions about their capabilities. This section covers Squid for caching web pages and related tasks, SquidGuard for filtering and controlling access to web content, and Lightsquid for reporting user activity based on the Squid access logs.

This discussion assumes the firewall running pfSense® software has a simple single LAN and single WAN configuration. Visit *Cache / Proxy* and Netgate Forum for additional guidance.

See also:

pfSense Hangouts on Youtube to view the March 2014 hangout on Squid, SquidGuard, and Lightsquid.

Warning: HTTPS traffic cannot be transparently intercepted in nearly all cases, this section only covers transparently capturing HTTP traffic. See *Transparent Proxies and HTTP/HTTPS* for details.

34.19.1 Squid Caching Web Proxy

Squid is the foundation of many other tasks that start with a proxy: It can act as a cache for improving web performance, it can hook into SquidGuard for content filtering, and its logs provide the basis for reporting on where users are going on the web.

Before anything else, the Squid package must be installed. Once installed, the package must be configured. The Squid configuration is broken up into several tabs. Before leaving a tab, click **Save**.

To start the configuration, navigate to **Services > Squid Proxy Server**.

Configure the Squid settings as follows, starting with the **General** tab:

Enable Squid Proxy *Checked*

Keep Settings *Checked*

Proxy Interfaces *LAN*, *loopback*

Allow users on Interface Checked, so that LAN users will be allowed to use the proxy.

Transparent HTTP Proxy Checked, so client HTTP traffic will be intercepted.

Bypass proxy for Private Address Destination Checked, so that local and VPN traffic will bypass the proxy.

Bypass Proxy for these Source IPs If certain local client IP addresses need to bypass the proxy, put them in this box. Multiple addresses, networks, or alias names may be entered separated by a semicolon.

Bypass Proxy for these Destination IPs If certain remote servers need to bypass the proxy, put them in this box. Multiple addresses, networks, or alias names may be entered separated by a semicolon.

Enable Access Logging To enable web access reporting, check this box.

Visible Hostname Enter the hostname of the firewall as presented to clients in proxy error messages.

Administrator E-mail Enter a usable contact address. If a user encounters a proxy error, this will be shown to the user so they may contact the address for support.

Save settings, then change to the **Local Cache** tab and configure it as follows:

Hard Disk Cache Size Set this to a value that is reasonable for the available drive space and RAM. If running with `/var` in RAM, enter `0` here.

Hard Disk Cache System If running with `/var` in RAM, set this to `null`

Other parameters on this tab can be tweaked as needed to control the size of objects to be cached, how much memory can be used for caching, and other related settings. Save the settings before navigating away from the page.

If there are more local subnets behind a static route on the LAN, visit the **Access Control** tab and add them into the **Allowed Subnets** list.

After these configuration steps have been completed, the proxy will be up and running. If transparent mode is in use, loading a proxy test site such as http://www.lagado.com/proxy-test will now reveal that the request was routed through a proxy.

34.19.2 SquidGuard Web Access Control and Filtering

The SquidGuard package enables very powerful URL content filtering and access control. It can use blacklists or custom lists of web sites, and can selectively allow or deny access to those sites. SquidGuard is capable of much more than will be covered in this section. Visit *Cache / Proxy* and the Netgate Forum for more information and related tutorials.

To use SquidGuard:

- Install and configure Squid as described in the previous section
- Install the SquidGuard package
- Navigate to **Services > Proxy Filter** to configure SquidGuard.

General Settings

- Navigate to **Services > SquidGuard Proxy Filter**, **General Settings** tab
- Check **Enable** to enable SquidGuard
- Click **Save**
- Check boxes to optionally enable other desired features, such as block event logging and GUI event logging

---

Note: After saving the settings on any tab in SquidGuard, always return to the **General Settings** tab and click the  **Apply** button. Until that action has been taken, the new SquidGuard settings will not be used.

Blacklists

Blacklists are predefined lists of sites in specific categories, such as **Social sites**, **Adult sites**, **Music sites**, and **Sports sites**. To use blacklists, check **Blacklist** and fill in a **Blacklist URL**. The two most common lists are the MESD list and the Shalla list.

Before the blacklist may be used, it must be downloaded and unpacked. To do this, after saving the settings on this tab, visit the **Blacklist** tab and click  **Download**.

Warning: If only blacklists are used, SquidGuard may fail. Define at least one **Target Category** as detailed in *Target Categories*.

Target Categories

Target Categories are custom lists of sites or other expressions that define a group of items that can be used to allow or deny access. They are maintained on the **Target Categories** tab.

When adding a new Target Category, a few options are required:

Name The Name for the category, as it will appear for selection on ACLs. The name must have between 2 and 15 alphanumeric characters, and the first character must be a letter.

Domain List This is the list of domain names to block, such as `www.facebook.com`, `google.com`, `microsoft.com`, etc. Multiple domains may be entered, separated by a space.

Redirect mode This option controls what happens when a user is blocked by a site in this list. The default of *none* will not redirect the user. The most common setting is *int error page*.

Redirect If the user is redirected using *int error page*, enter the error message that will be presented to the user here. If an external redirect type is used, enter the full URL to the desired target site, including the proper protocol such as `http://` or `https://`.

Access Lists (ACLs)

There are two types of ACL entries in SquidGuard:

1. Common ACL, which is the default ACL applied to all users
2. Group ACL entries which are applied to specific IP addresses, groups of IP addresses, or Networks.

First, visit the **Common ACL** tab. Choose the default actions for all available categories from blacklists or those defined locally. To do this, click **Target Rules List**  , and pick the desired actions from the drop-down at the end of the row for each category. The **Default Access [all]** choice controls what happens when no match has been found in any of the available categories.

After saving the settings, change to the **Group ACL** tab to create an entry for a specific user or group of users. Using a Group ACL, an exception to the Common ACL rules may be crafted, either to block access to a site others can reach, or to allow access to a site that others are blocked from viewing.

To create a **Group ACL**:

- Change to the **Group ACL** tab
- Click  **Add** to start a new entry and configure it as follows:

  Name The name of the ACL

  Client (source) Enter the user's IP address, subnet, etc. Multiple values can be entered, separated by spaces.

  Target Rules List Defines the list of actions for this specific set of users
- Click **Save**
- Return to the **General Settings** tab

- Click **Apply**

34.19.3 Lightsquid Web Access Reporting

Lightsquid is used to create reports that detail the web history of computers that have accessed sites through the proxy. After the Lightsquid package has been installed, the report settings may be found under **Status > Squid Proxy Reports**.

The look and feel of the reports may be customized by choosing the **Language**, **Bar color**, and **Report Scheme**.

The **Refresh scheduler** option controls how often the report will be automatically updated, e.g. every *30* minutes.

Click **Save** to store the settings and then click  **Refresh Full** to build the initial report. Wait a few minutes, then click the  **Open Lightsquid** button to view the report.

If there is no data in the report, check to make sure that **Enable Logging** is set in Squid, and that the user traffic is going through the proxy as expected.

34.19.4 Transparent Proxies and HTTP/HTTPS

When using a proxy, it is only possible to intercept **HTTP** traffic transparently. That is, only HTTP traffic may be grabbed automatically and forced through a proxy without intervention from the user or their knowledge. This is convenient, since it does not require configuring any settings on the user's PC. The downside is that **only HTTP** traffic may be captured using this method; It is not possible to intercept HTTPS in the same way.

Attempting to transparently intercept HTTPS would break the chain of trust created by SSL, causing the user to be greeted with a scary certificate warning when they attempt to access a secure site. This warning would be valid in that case, because the proxy is essentially performing a man-in-the-middle attack in order to inspect the user's traffic.

The Squid proxy package is capable of intercepting HTTPS, but it cannot be done completely without the knowledge of the user or alterations to their computer. At a minimum, intercepting HTTPS requires the installation of a trusted root CA that has been created for this purpose, so that the proxy can appear to use valid certificates.

The best method is to place the proxy settings into the user's computer and/or browser software. This task can be done manually, via GPO on a Windows Domain, by DHCP, or automatically using WPAD. The details of those are beyond the scope of this documentation, but there is information on many of those tactics on *Cache / Proxy* and Netgate Forum.

34.20 Authenticating Squid Package Users with FreeRADIUS

Using the Squid Proxy package with the *FreeRADIUS package*.

34.20.1 SQUID Proxy

Squid provides the possibility to ask for a username and password for users who want to connect to the internet through squid proxy. This works only if squid is running in non-transparent mode.

- SQUID configuration:
  - Disable transparent mode in **Proxy Server > General**
  - Enable RADIUS as authentication method in **Proxy Server > Auth Settings**
- FreeRADIUS configuration:
  - Configure an interface in **FreeRADIUS > Interfaces**
  - Configure a user in **FreeRADIUS > Users**
  - Configure a NAS/Client in **FreeRADIUS > NAS/Clients**. In this case pfSense® software is the NAS/Client. So enter the pfSense IP-Address.

For squid in non-transparent mode the IP address and the squid port must be entered on the host's browser. When a user connects to the Internet through the proxy then the browser will present a login window where the user has to enter username and password.

34.21 Configuring the Squid Package as a Transparent HTTP Proxy

This recipe describes how to install and configure Squid as a transparent proxy on pfSense® software.

34.21.1 Install the Package

First, install the Squid package.

1. Click **System > Package Manager**
2. Click **Available Packages**
3. Enter `squid` in the search bar and click search or scroll down until the **squid** package listing is visible
4. Click the **install** button on the far right
5. Click **Confirm** when prompted ("Confirmation Required to install package pfSense-pkg-squid")
6. Wait for the installer to download, install, and do post-install tasks for squid, such as creating the cache directories.

34.21.2 Configure the Squid Package

After the installation has finished, the Squid proxy server may be configured.

1. Click on the **Local Cache** tab.
   1. Hard disk cache size (in MB): Set this as needed, but keep it a reasonable size. 3000 (3GB) may be a good place to start.
   2. Hard disk cache location: Should be `/var/squid/cache` but may be moved if needed
   3. Memory cache size: The amount of RAM that squid should claim for caching. Use as much as can be spared, as this is much faster than caching to disk. It should not exceed 50% of the installed RAM, however.

4. Hard disk cache location: The directory where the cache will be stored. If using a non-default location enter it here.
5. Minimum object size: Can be left at 0 to cache everything, but may be raised if small objects are not desired in the cache.
6. Maximum object size: Objects larger than this setting will not be saved on disk. If speed is more desirable than saving bandwidth, this should be set to a low value.
7. Do Not Cache: Set a list of domains that should never be cached. This may also be left blank.
8. Click **Save**.

2. Click on **Services > Squid Proxy Server**
3. Set the options on the **General** tab as desired.
    1. Proxy Interface(s): Select which interface(s) the proxy will **listen** on. *LAN* is probably the desired setting.
    2. Allow users on interface: If this is checked, the subnets for the interfaces selected in the last step will automatically have access. There will be no need to add them on the Access Control tab.
    3. Transparent Proxy: Check this to have pfSense software automatically redirect outbound HTTP (tcp/80) traffic through the proxy.
    4. Enabled logging: Check this if logging is needed, be sure to put a path in the following box
    5. Log Store Directory: Should be `/var/squid/log` unless another location is absolutely necessary.
    6. Proxy Port: **Leave this as 3128**. There is no need to change the port number for the transparent proxy to work.
    7. The remaining settings may be left at their defaults, or changed if desired. It is likely best to leave them alone until the proxy is operational and tested.
    8. Click **Save**.
4. Click on the **ACLs** tab *(optional for most)*
    1. If any other subnets will pass through the proxy aside from the subnet for the interface squid is using, enter them here.
    2. Click **Save**.

That's it! Squid should be up and running. The status of the squid proxy can be checked by clicking **Status > Services**.

Also available are:

- **Lightsquid** package to view web access reports from the squid log.
- **squidGuard** package for who wish to have more fine-grained control over what web resources may be viewed by clients.
- *Squid Package Tuning*

34.22 Setting up WPAD Autoconfigure for the Squid Package

pfSense® software can be configured to serve up automatic proxy configuration data to clients to point users to squid running either on the pfSense system or another local system, assuming their systems settings are configured for this behavior. Though the data can be served from the firewall, the task is better suited for another local web server if one is available.

Note: To use the web server on the firewall to serve this data, the GUI must run in HTTP mode, or the vhosts package may be used to setup an alternate HTTP server on port 80. Neither of these are recommended as much as running a separate local web server for this task.

This process is known as WPAD, short for Web Proxy AutoDiscovery Protocol. If a web browser is configured for autodiscovery, it will try a few methods to figure out a proxy's location.

A WPAD host may be supplied via DHCP numbered option 252 (string value containing the entire URL to the WPAD file) or DNS, which is easy to do with the built-in DNS forwarder.

34.22.1 Why would this be done?

To use squid authentication, squid cannot be used in transparent mode. HTTPS traffic also cannot be filtered using transparent mode. When squid is run in normal mode, a proxy IP and port must be configured on each client machine, which can be tedious. This can also cause problems on road warrior laptops that come in and out of the network. Rather than resetting their proxy configuration each time they enter and leave, autoconfigure will let them come and go without much trouble.

Most, if not all, modern browsers ship with the autoconfigure setting turned off, so it may still be necessary to push/enter this setting to client PCs. Even so, another advantage of using autoconfigure is that should squid move to another IP address, only one file must be changed to inform the clients of the updated IP address. (This may be easy to pull off in a windows domain with AD, but not for many others!)

34.22.2 Prerequisites

This recipe assumes squid is already operating in a non-transparent configuration. For help with that, look in *A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid* and on the Netgate Forum.

34.22.3 Create wpad.dat

Before starting, a *wpad.dat* file must be crafted. This is a single file with a JavaScript function which tells the browser how to find a proxy hostname and port. This function can be as simple or as complex as desired, there are many examples on the web. In this example, all clients will be directed to the squid instance on the firewall.

The contents of the example *wpad.dat* file are:

```
function FindProxyForURL(url,host)
{
return "PROXY 192.168.1.1:3128";
}
```

The function in that file tells the browser to look for a proxy on *192.168.1.1* at port *3128*.

Now upload that file to the pfSense system or another locally accessible web server with scp, or create it using the built-in file editor. The file must go in */usr/local/www/*.

Due to the different ways that various browser versions try to access the file, this same code should exist in at least three different places:

```
/usr/local/www/wpad.dat
/usr/local/www/wpad.da
/usr/local/www/proxy.pac
```

(More advanced users might do this from the shell and use *ln* to link the files.)

We recommend pointing *wpad.* to an internal web server which can answer requests for the *wpad.dat* and associated files. It can be any web server, but typically must be served from both the default *VirtualHost* as well as one named *wpad*, due to differences in how browsers request the file.

To make this work using the pfSense router to serve the file, local IP addresses will need to be able to access the local interface IP address of the pfSense router. They do not need to access the WebGUI with a password, this file will be served without authentication. The GUI must also be run in HTTP mode, which is less secure. If the GUI is set to use HTTP, never open up access to the GUI over the WAN.

34.22.4 Configure DNS

Now to setup the DNS portion. WPAD will take the domain name given to the machine, likely assigned by DHCP, and prepend *wpad.*. If the domain is *example.com*, it will look for *wpad.example.com*. This task may be accomplished with the DNS Forwarder/DNS Resolver on the pfSense router or with another internal DNS server used by client PCs.

A client browser will ultimately try to access *http://wpad.example.com/wpad.dat* - among others. More details on the hostnames tried by WPAD are available in the WPAD article on Wikipedia.

To add the entry using the DNS forwarder on the pfSense router, navigate to **Services > DNS Forwarder**. Click  to add a new **Host Override**.

Enter the following (Replace the domain and IP address with their actual values):

- **Host**: *wpad*
- **Domain**: *example.com*
- **IP Address**: *192.168.1.1*
- **Description**: *WPAD Autoconfigure Host*

Click **Save**.

34.22.5 Block Port 80 Out from LAN

Create a firewall rule at the TOP of the LAN tab (or appropriate interface) that blocks anything from **<internal subnet>** to * on port 80.

Note: If the firewall is used to serve WPAD and the WebGUI anti-lockout rule has been disabled, web traffic must also be allowed to the pfSense firewall GUI port. If this is not acceptable, point *wpad.* to another internal web server which can answer requests for the *wpad.dat* and associated files.

34.22.6 Test Clients

Start a browser on a client behind the pfSense firewall, and see what happens. If squid is configured for authentication, the client will be greeted with a login prompt. Otherwise, check squid's logs to ensure traffic is going through the proxy. A proxy test site such as http://www.lagado.com/proxy-test can also be useful.

If nothing happened, check the browser settings. Many modern browsers ship with the autoconfigure settings off.

Internet Explorer

- Open **Internet Options**
- Click the **Connections** tab
- Click the **LAN Settings** button
- Check **Automatically Detect Settings**
- Click **OK**, and **OK** again.

Firefox

- Click **Tools** (Or the three bar icon)
- Click **Options**
- Click **Advanced**
- Click the **Network** tab
- Click the **Settings** button
- Select **Auto-detect proxy settings for this network**
- Click **OK**

34.23 IPsec Remote Access VPN Example Using IKEv1 with Pre-Shared Keys

This article describes how to set up Mobile IPsec in pfSense® software with a Pre-Shared Key, and how to configure the Shrew Soft VPN Client to match. The Shrew Soft VPN client is freely available for Windows, Linux and BSD at http://shrew.net.

Note: The current best practice is to use IKEv2 for IPsec Remote Access on modern clients. See *IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2* for details.

Prerequisites:

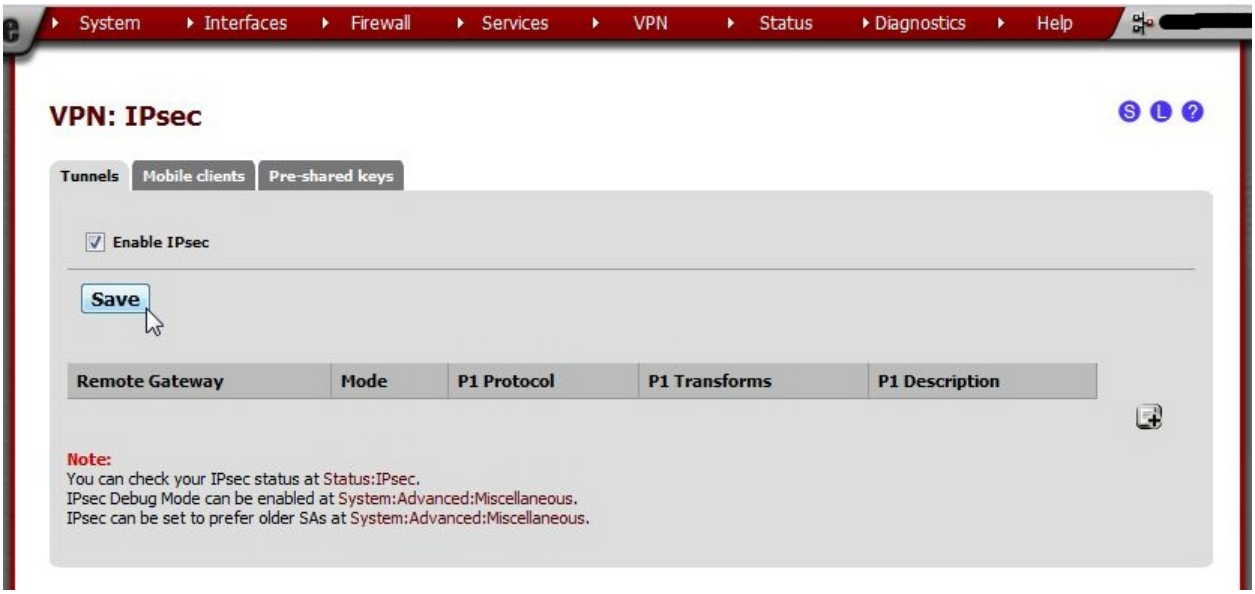
- Make sure the LAN is using the firewall as its default gateway and that it is working properly.
- Make sure the client has a functioning Internet connection.

If either condition is not met, the tunnel will not work.

34.23.1 On pfSense

Begin by enabling IPsec.

- Navigate to **VPN > IPsec**
- Check **Enable IPsec**
- Click **Save**



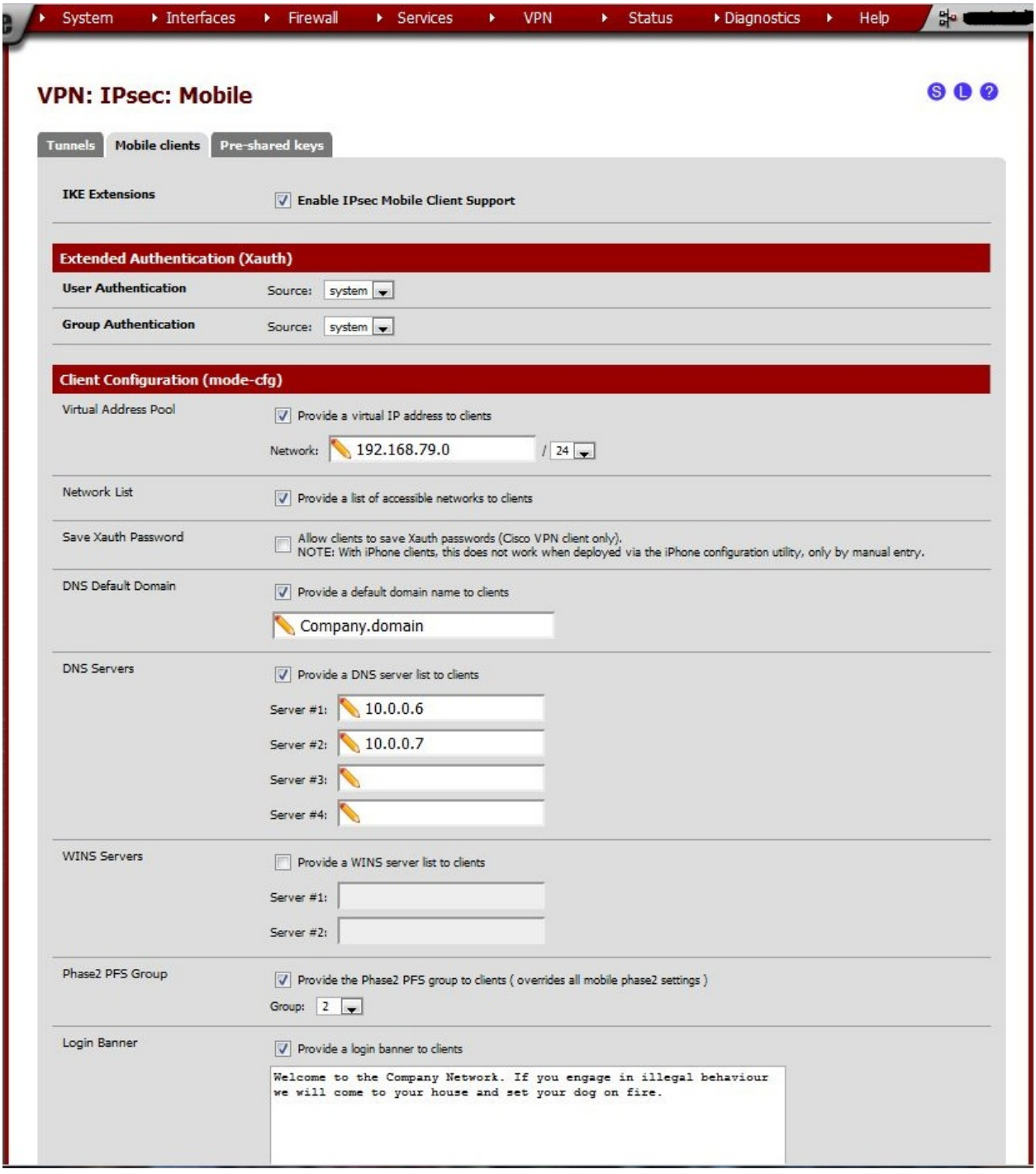
Now, create a phase 1 entry.

Do *not* click  on this page to create a phase 1 entry. That button will not go the page needed to create a phase 1 for mobile clients but will go to a page to create a phase 1 for lan-to-lan-tunneling instead.

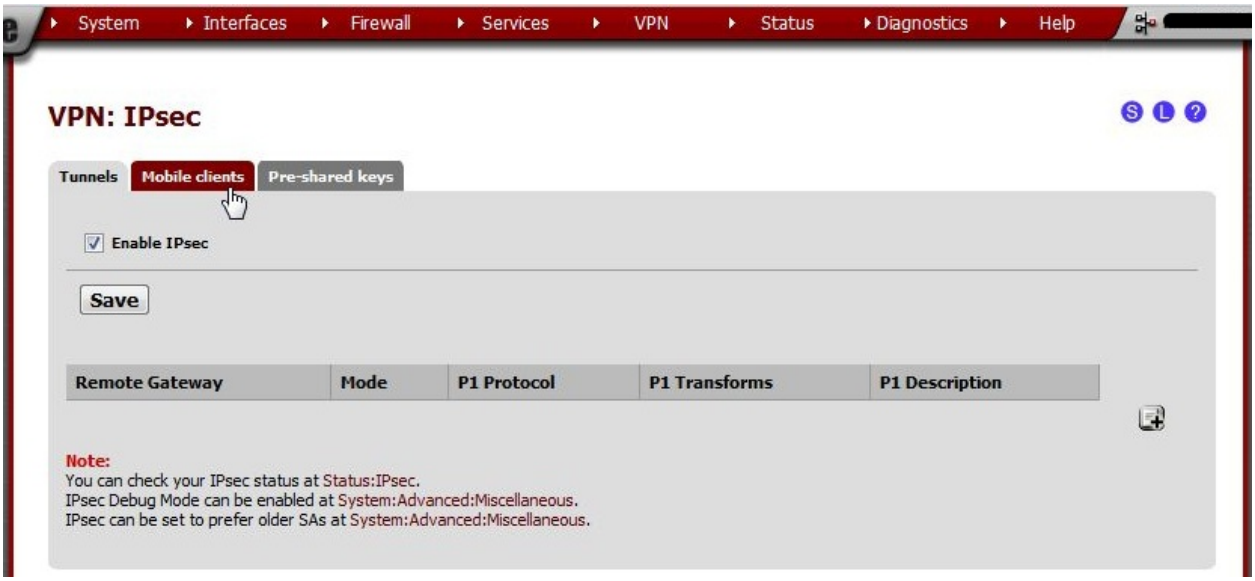
Navigate to the **Mobile clients** tab. Check **Enable IPsec Mobile Client Support**

Tell the client about available services. The more items entered here, the less clients have to enter manually. Enter the following values.

Key	Value	Remark
IKE Extensions	checked	
User Authentication	system	
Group Authentication	system	
Virtual Address Pool	checked, network: 192.168.79.0/24	Enter a network here that is not in use in the LAN and preferably not in any clients' LAN either. It can be any subnet, but don't pick a commonly used one (e.g. don't use 192.168.0.0/24 or 192.168.1.0/24). It will confuse the clients.
Network List	checked	
Save Xauth Password	unchecked	Check if using xauth and the clients should be able to save passwords locally.
DNS Default Domain	Check to supply a default DNS domain for hostname resolution by clients	Optional but if a domain is present (such as Active Directory) clients will be able to resolve servers faster.
DNS Servers	Check if clients must get DNS over IPsec	If Active Directory is used, enter its DNS servers here. If it's a home network, the IP address of the firewall, Google public DNS, OpenDNS, or any other DNS server reachable via the VPN may be used.
WINS Servers	Check if WINS is in use	Superfluous if also providing DNS but may be needed for some older domain configurations.
Phase2 PFS Group	checked, group 2	Not necessary to set here, as it may be set in the Phase 2 of the mobile IPsec tunnel settings
Login Banner	Optional	Client software which honors the login banner will present this text to the user upon login. May be needed to display some legal information or any other welcome message.



When finished, click **Save**, then click **Apply Changes**.



After saving, a warning will appear **Support for IPsec Mobile clients is enabled but a Phase1 definition was not found. Please click Create to define one.**

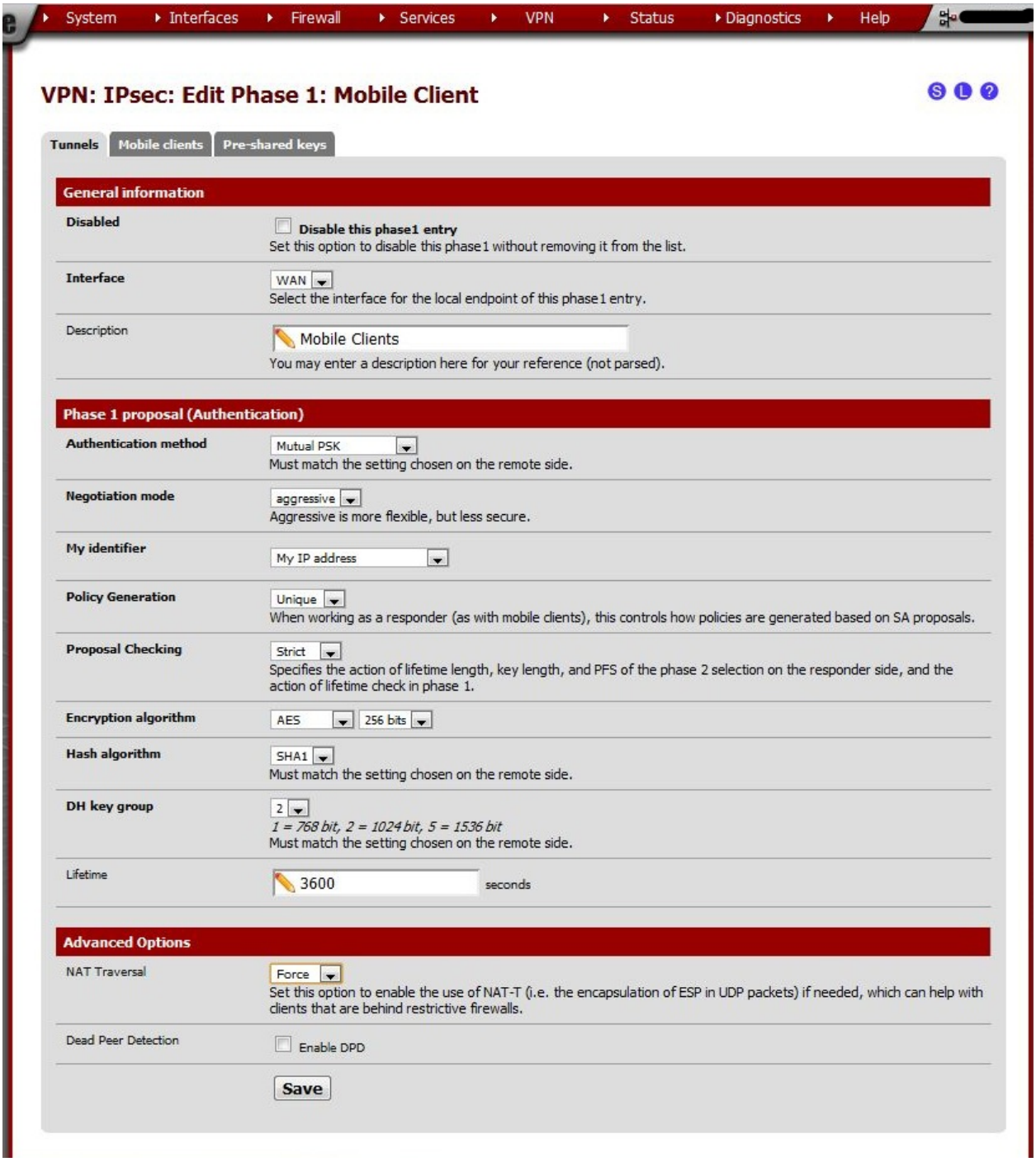
Click the **Create Phase1** button.



Clicking the button will load the appropriate page to create a Phase 1 for mobile clients.

On the **VPN: IPsec: Edit Phase 1: Mobile Client** page, enter the following values:

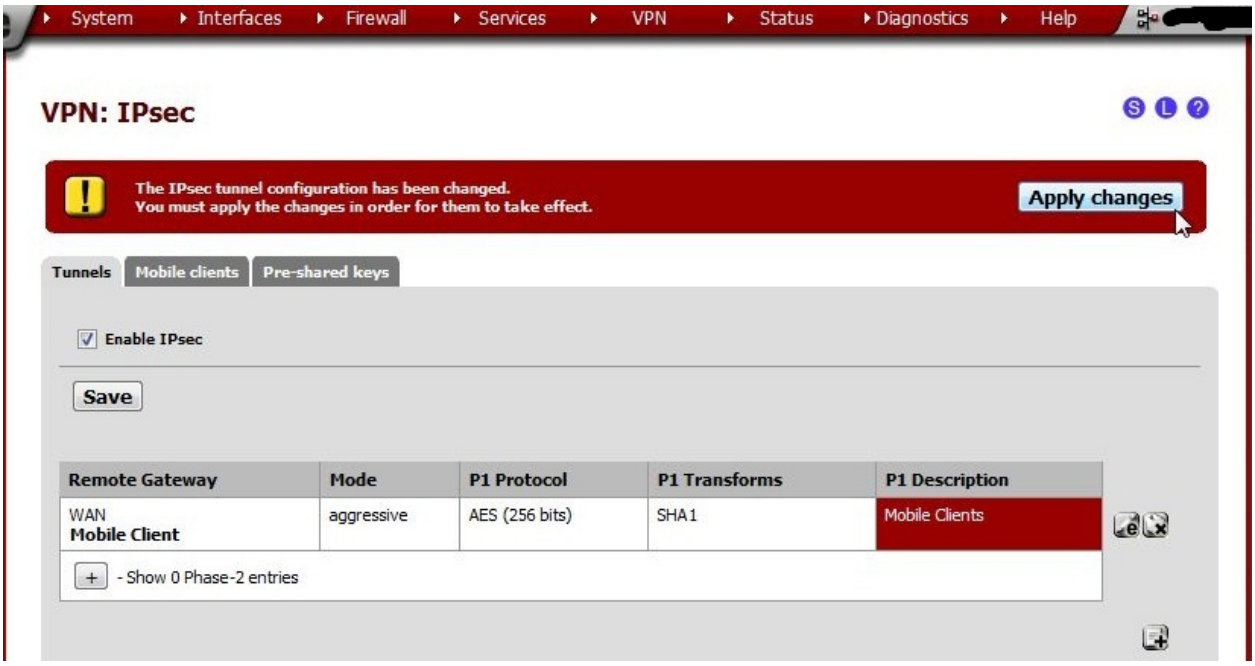
Key	Value	Remark
Disabled	not checked	
Interface	WAN	
Description	Mobile Clients	This can be anything, name it something appropriate.
Authentication method	Mutual PSK	
Negotiation mode	aggressive	
My identifier	My IP address	
Policy Generation	Unique	Might prevent traffic to the LAN if set to something else.
Proposal Checking	Strict	
Encryption algorithm	AES, 256 bits	Choose any, but keep it identical on router and client.
Hash algorithm	SHA1	
DH key group	2	
Lifetime	3600	
NAT Traversal	Force	Might prevent traffic to the LAN if set to something else.
Dead Peer Detection	not checked	



Click **Save**.

A warning will appear: **The IPsec tunnel configuration has been changed. Apply the changes for them to take effect**.

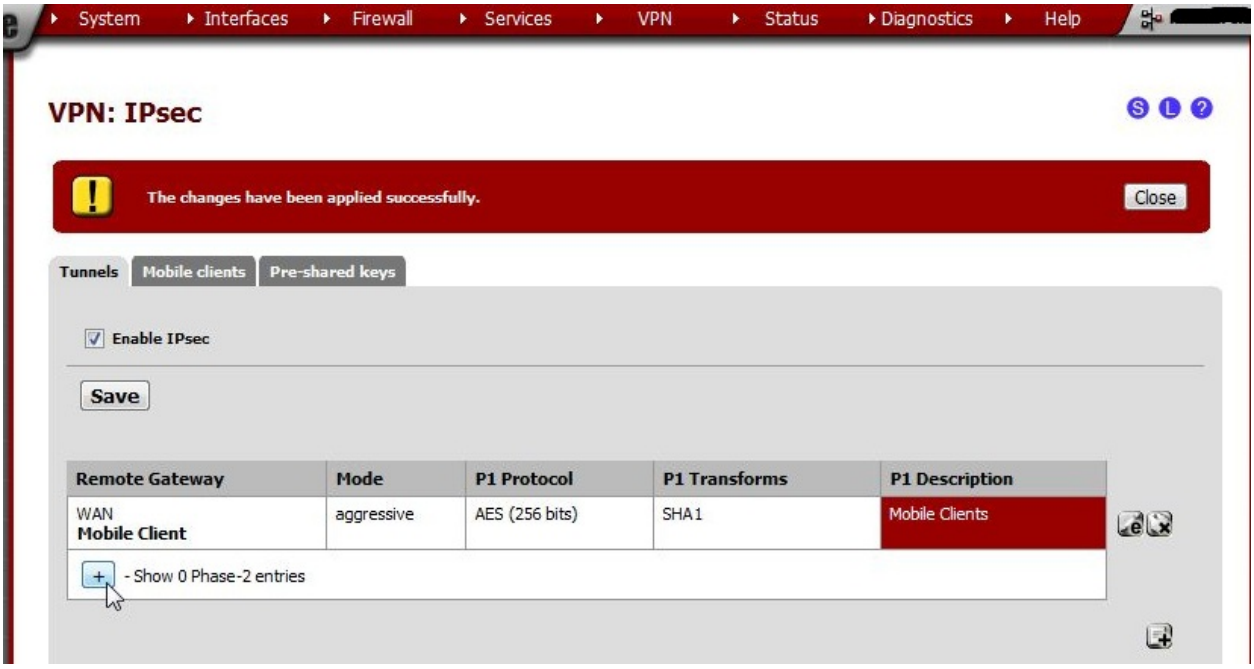
Click **Apply changes**.



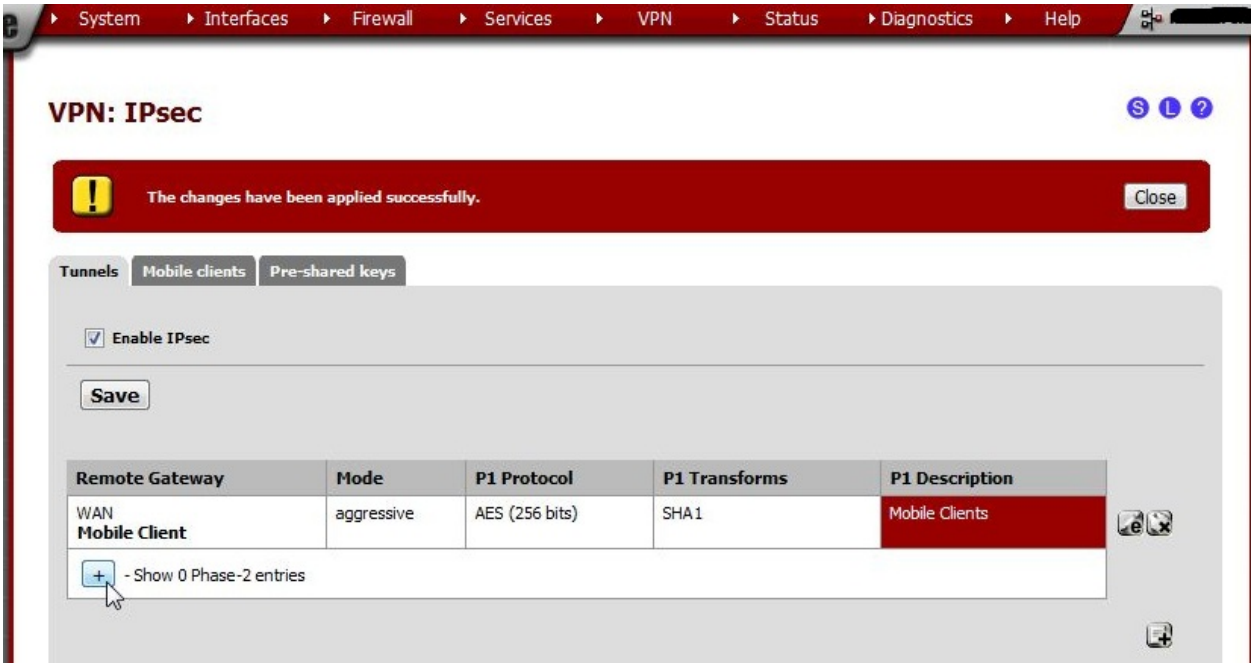
The notice for **The changes have been applied successfully.** may be ignored. The neurotics among us may click the **Close** button but that's optional.

With phase 1 created, we can create a phase 2.

Click  to list the Phase 2 entries under the newly created Phase 1.



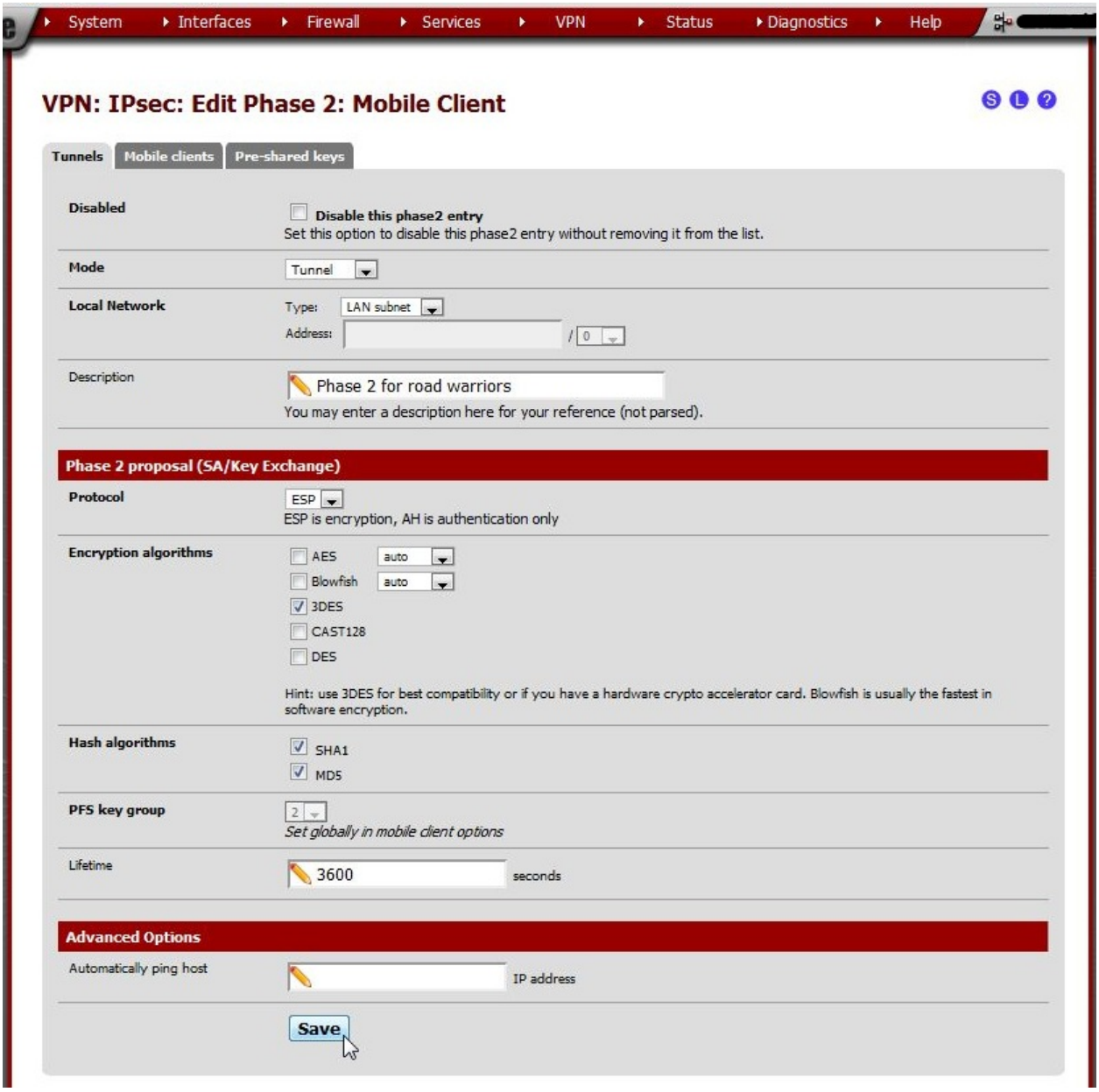
Surprise! There aren't any. Create one by clicking  in the Phase 2 list.



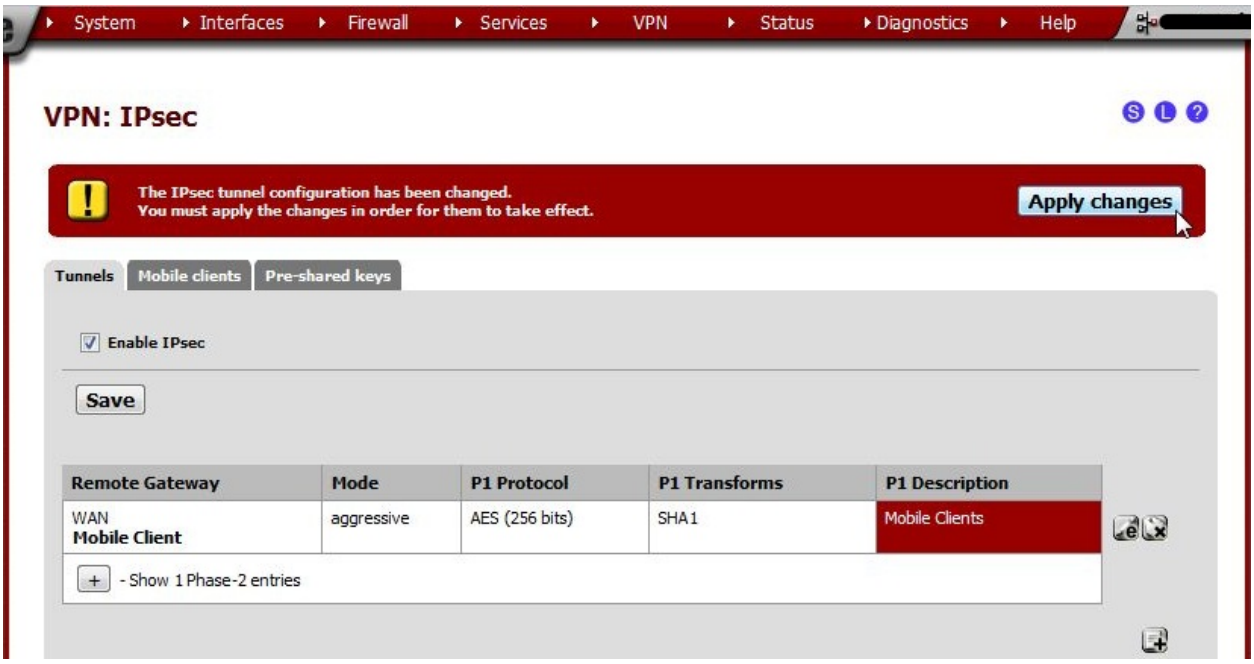
This will open the *VPN: IPsec: Edit Phase 2: Mobile Client* page.

On the **VPN: IPsec: Edit Phase 2: Mobile Client** page, enter these values:

Key	Value	Remark
Disabled	not checked	
Mode	Tunnel	
Local Network	LAN subnet	
Description	Phase 2 for road warriors	Enter something appropriate.
Protocol	ESP	
Encryption algorithms	select only 3DES	The best is chosen at handshake time. Others will probably work too. 3DES works for me because I have a mobile application that will work only with this.
Hash algorithms	Select SHA1 and MD5	
PFS key group	Set to Group 2	
Lifetime	3600	
Automatically ping host	leave empty	



Click **Save**, then click **Apply changes**.



We're almost done here. We need to create user accounts so someone can actually use the tunnel.

Navigate to **VPN > IPsec**, **Pre-shared keys** tab. (Screenshots may look a bit different because in-use keys have been redacted.)

There are different ways to set up pre-shared keys for users. They may also be added in the *User Manager* but that is beyond the scope of this document.

Click  to create a new Pre-Shared Key.

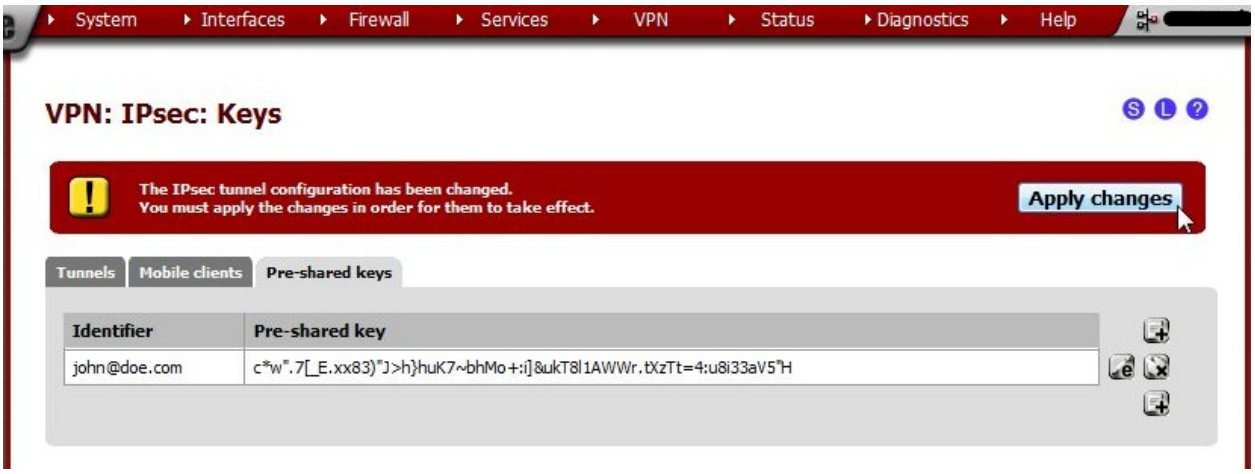


For identifiers, e-mail addresses are commonly used as they are more unique than first or last names. Any identifier may be used so long as it is unique to the person using the account. We recommend using e-mail address format identifiers. They don't really need to exist, they are only used for IPsec identification.

Generate a long/random Pre-Shared Key. There are many utilities to generate random data, such as Lastpass, KeyPass, or online sites such as https://www.grc.com/passwords.htm. (Use the string in the middle: 63 random printable ASCII characters). Offline tools are preferred.



Press **Save**, wait for the page to load, note that the key is now in the list and press **Apply changes**.



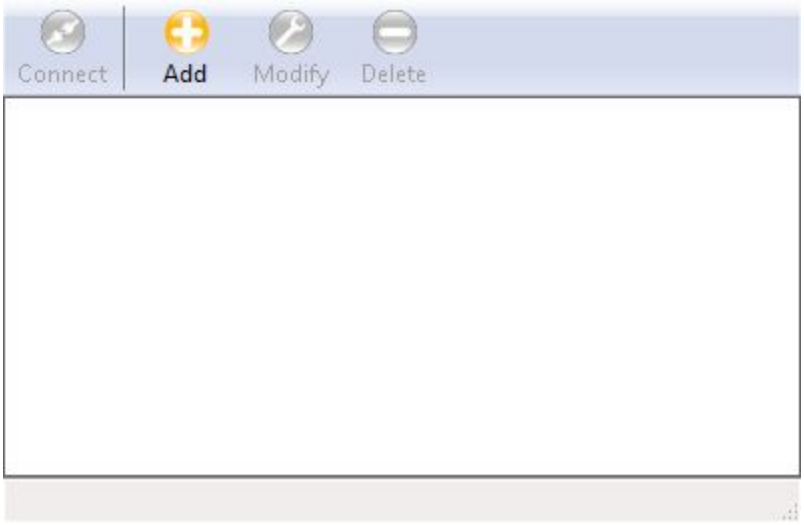
Congratulations, the firewall configuration is complete.

34.23.2 The client

This part is done on the user's computer. Screenshots were taken in Windows but Shrew Soft VPN is available for Linux and BSD (so probably Mac) too.

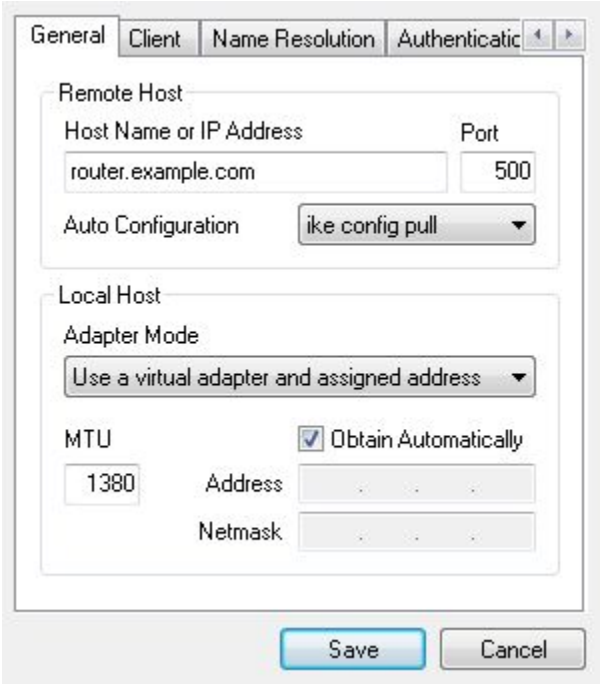
Download and install Shrew Soft VPN.

Once finished, open ipseca.exe. The VPN Access Manager window is presented. (Window title bar is missing in the screenshots)

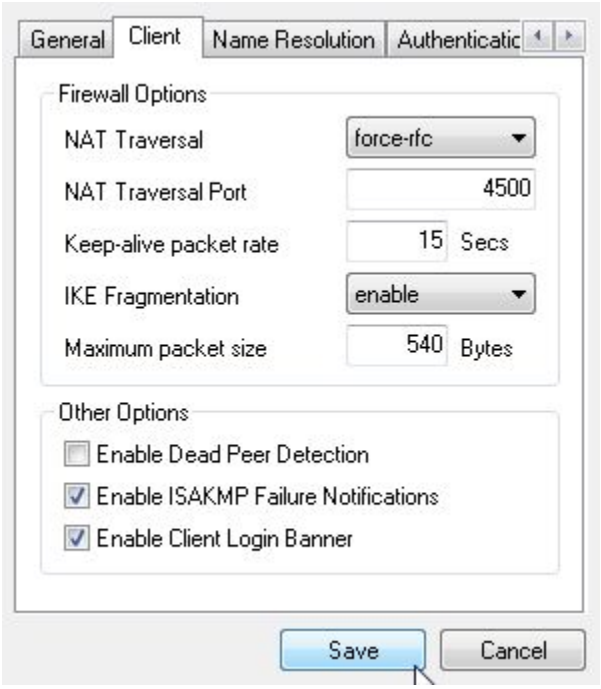


Press the big round **Add** button to set up a tunnel configuration.

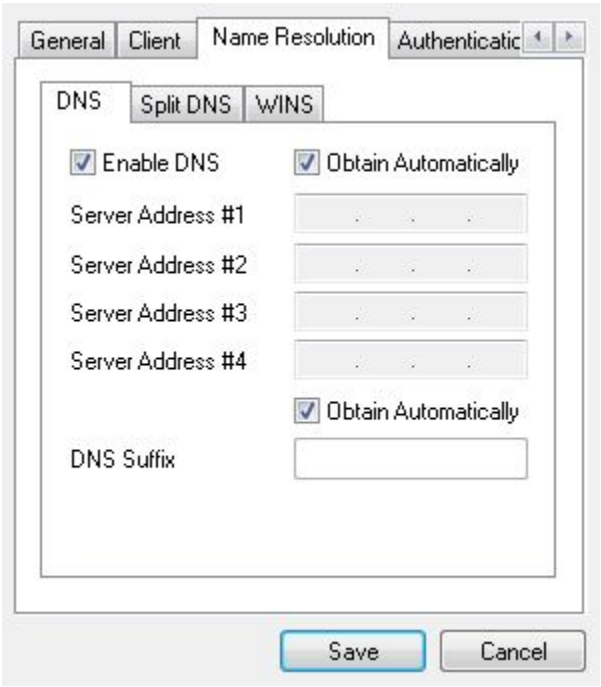
On the **General** tab, enter the IP address or host name pfSense firewall. Leave the rest as it is. The default values in new versions of the Shrew Soft VPN client may change so in case of doubt, stick to the screenshots.

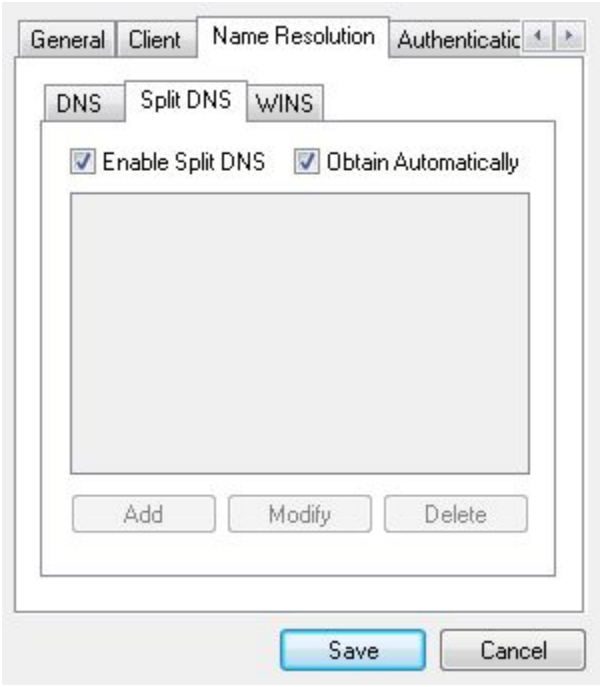


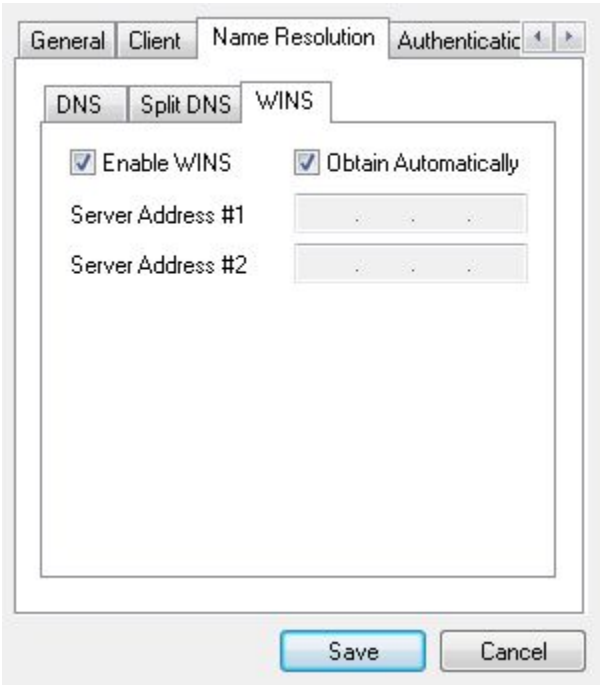
On the **Client** tab, set **NAT Traversal** to **force-rfc** and uncheck **Enable Dead Peer Detection**. If these settings are wrong, an established tunnel may not let any traffic through.



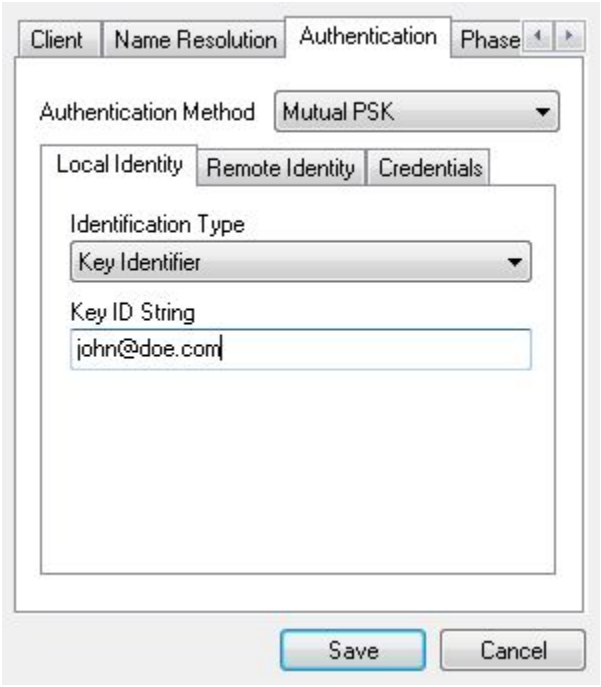
Don't change anything on the **Name Resolution** tab; these settings are all automatically set by the pfSense software. Relevant information could be entered here but if the settings were configured on the firewall, they need not be set here.



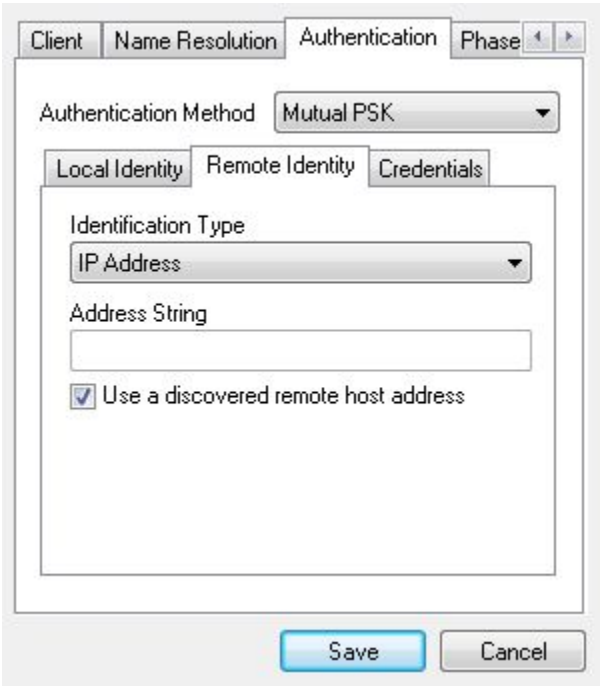




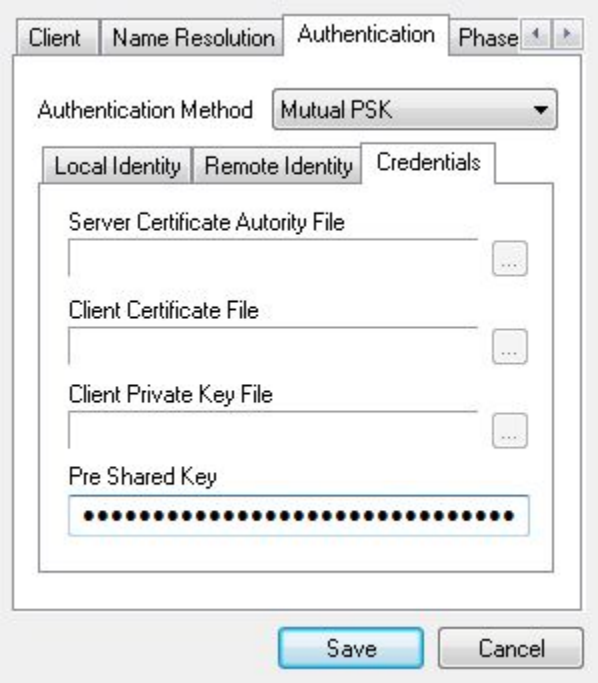
Go to the **Authentication tab**. Set **Authentication Method** to **Mutual PSK**. Under **Local Identity**, choose **Key Identifier** as the **Identification Type** and enter the user's e-mail address (or whatever was used as an identifier) in the **Key ID String** field.



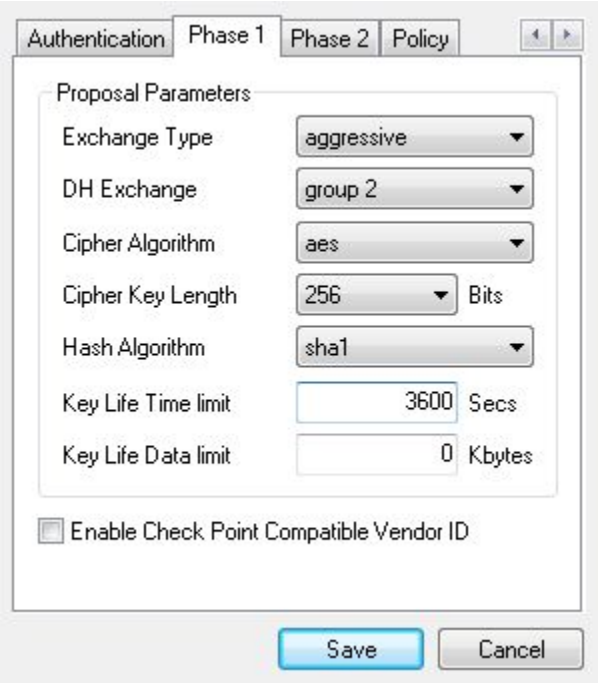
Under **Remote Identity**, set **Identification Type** to **IP Address** and check **Use a discovered remote host address**.



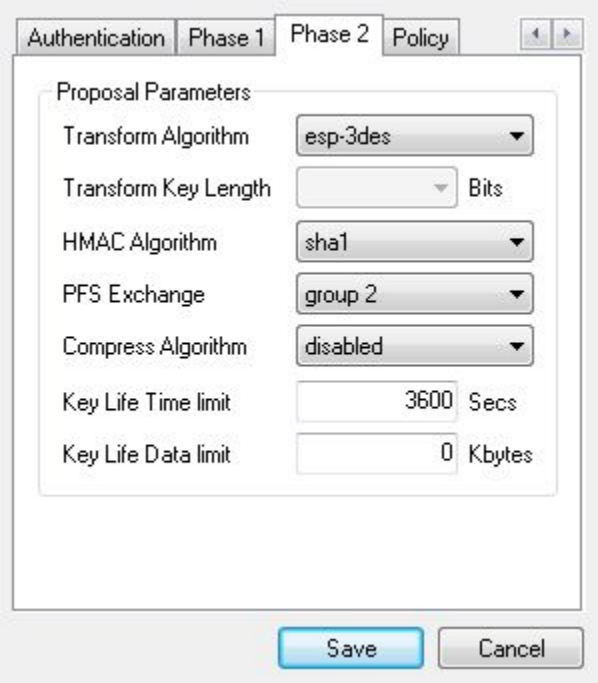
Finally, under **Credentials**, enter the Pre Shared Key associated with the e-mail address.



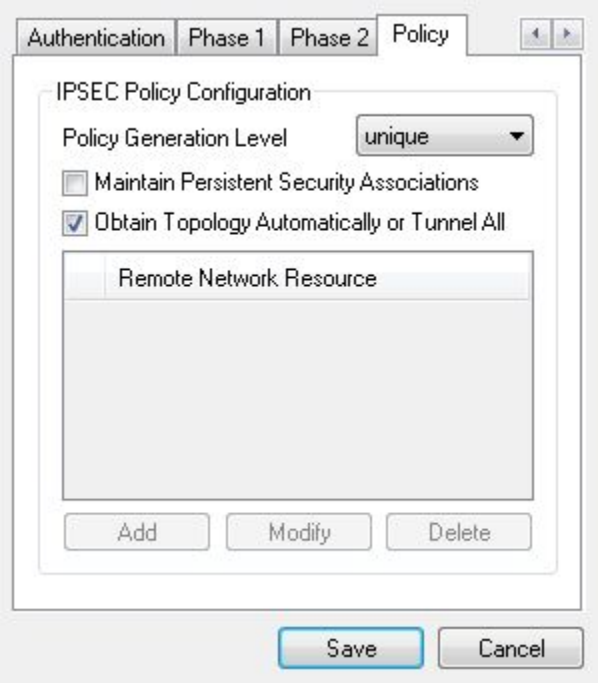
Now scroll over to the **Phase 1** tab. Set the **Cipher Algorithm** to **aes** or whatever was entered on the Phase 1 page in the pfSense software. **Cipher Key Length** to **256** (or whatever etc.) and **Hash Algorithm** to **sha1**. Set the **Key Life Time limit** to **3600**.



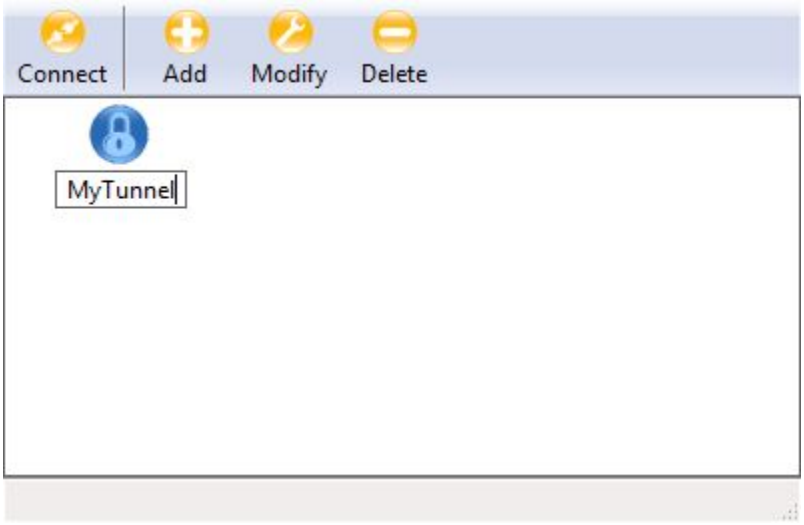
Phase 2 tab: set **Transform Algorithm** to **esp-3des**, **HMAC Algorithm** to **sha1** and **PFS Exchange** to **group 2**.



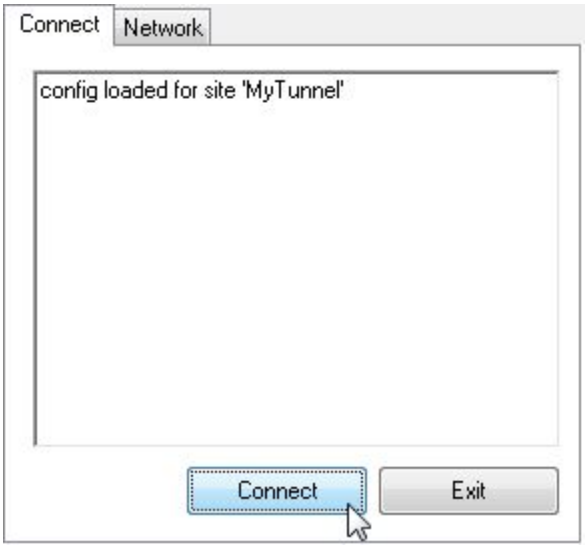
Nearly there! Go to the **Policy** tab and set **Policy Generation Level** to **unique**.



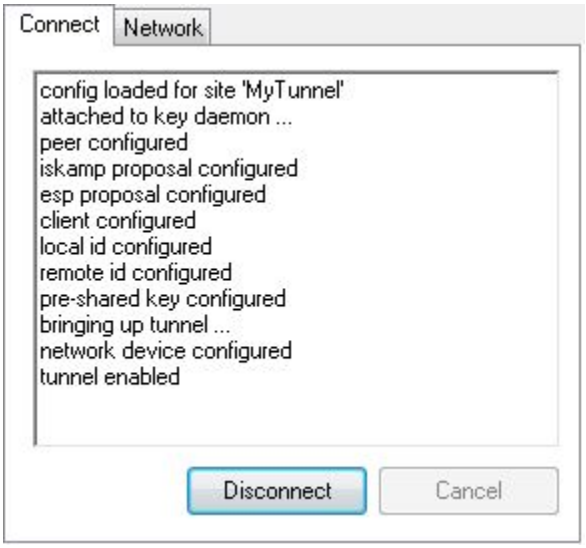
Click **Save** and give the newly created configuration an appropriate name.



Double-click the configuration and the tunnel window will pop up. Click **Connect** to start the tunnel.



Click **Disconnect** to... disconnect the tunnel.

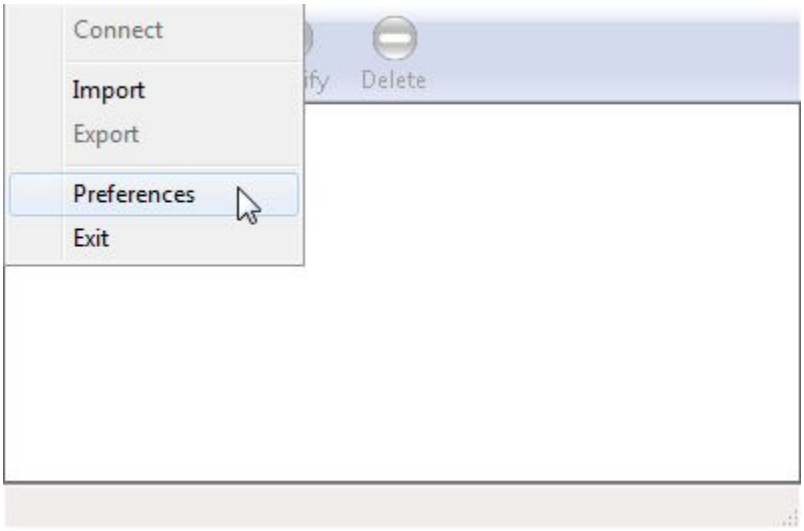


That's it! A working IPsec tunneling system is now in place.

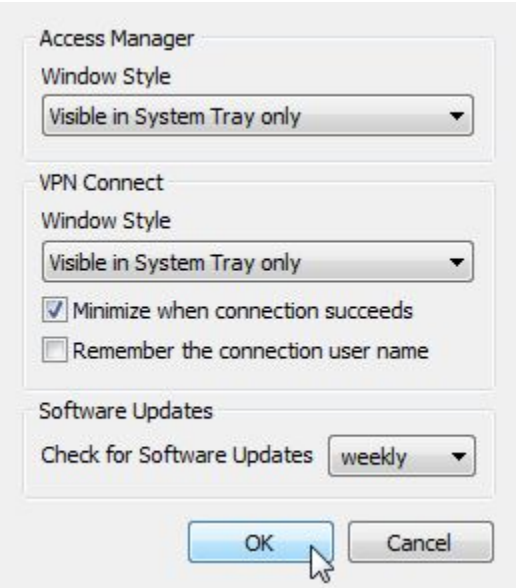
34.23.3 Client tweaks

Personally I like to tweak it a little bit so the windows hide themselves nicely in the system tray. This is optional but I find it improves the user experience.

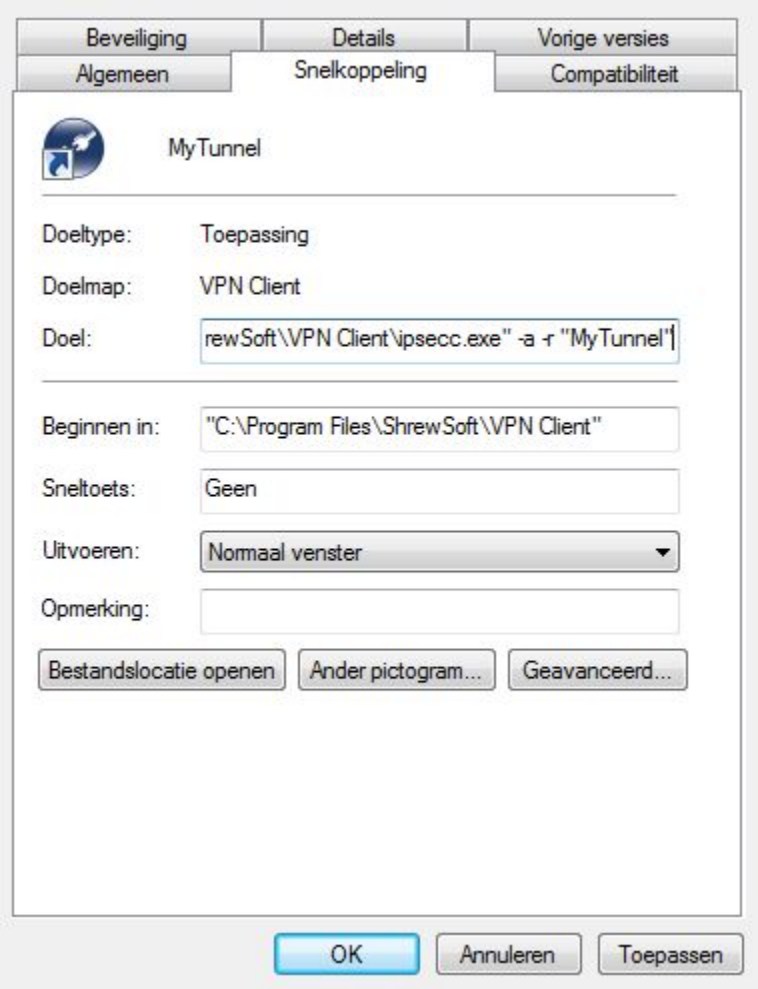
In the VPN Access Manager, go to **File > Preferences**.



For Access Manager and VPN Connect, set **Windows Style** to **Visible in System Tray only** and check **Remember when connection succeeds**. No need to remember the user name since we're not using user names but pre-shared keys.



A shortcut may be created directly to the tunnel: create a shortcut to ipsecc.exe (in `c:\program files` etc.). Right-click the shortcut and choose **Properties**. In the **Target** field, add **-a -r "MyTunnel"**. -a means: start automatically. This starts the connection without the user having to press the Connect button. -r specifies the tunnel name. If the tunnel was named "Work", write "Work" in stead of "MyTunnel".



Now when the shortcut is double clicked, the tunnel is automatically started.

Backup the tunnel profile by selecting it in the VPN Access Manager and going to **File > Export**. Restoring works by choosing **Import**.

34.23.4 Troubleshooting

I've been using pfSense software in combination with Shrew Soft VPN for a long time and in my experience it is a very stable combination. However things can always go wrong. If it doesn't work, here are some hints to help troubleshoot.

- Check the router and the client settings.
- Check the router and the client settings again.
- In the pfSense webGUI, go to **Status > System Logs** and there to the **IPsec** tab. Hit the **Clear log** button, have the client try and start the connection and click the **IPsec** tab again to refresh the page. This is usually very inspiring.
- Reboot the client machine.
- Reboot the pfSense machine. Should not be necessary but if all other attempts fail, it may be tried.
- Use a simple pre-shared key so mistake can be eliminated. When done troubleshooting, use the hard key again!

- If a user calls and says Shrew Soft VPN wants to know his user name and password, it's almost always because the user has either no Internet connection or no dns service. Or they are on a guest network and need to open their browser for identification or something.
- Roy Blüthgen wrote in to say: I am running a pfSense software version 2.0.2 installation and followed the guide to set up IPsec server/client. Afterwards when testing I was running into this issue: https://redmine.pfSense.org/issues/1351. I tried the pfSense config suggested in note 30 (by Jim) and that fixed my problem: **System > Advanced**, **Miscellaneous** tab, **IP Security** section: disable/uncheck **Prefer older IPsec SAs** (added this info as note 35 for issue 1351)

34.24 IPsec Remote Access VPN Example Using IKEv1 with Xauth

This document covers IPsec using Xauth and a mutual Pre-Shared Key.

Note: The current best practice is to use IKEv2 for IPsec Remote Access on modern clients. See *IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2* for details.

This setup has been tested and working on various *Android* and *iOS* devices. Other clients may work as well.

34.24.1 IPsec Server Setup

This is the setup for the pfSense® side of the connection

Mobile Clients

- Navigate to **VPN > IPsec**, **Mobile Clients** tab
- Check **Enable IPsec Mobile Client Support**
- Check **Provide a virtual IP address to clients**
- Enter an unused subnet in the box, pick a subnet mask
- Set any other desired options here
- Click **Save**
- Click **Apply Changes**
- Click **Create Phase1** (if it appears)

Phase 1 settings

- Navigate to **VPN > IPsec**
- Locate the Mobile Phase 1 in the list
- Click  to edit the Mobile Phase 1
- Enter the following settings:
  - **Authentication method**: *Mutual PSK + Xauth*
  - **Negotiation mode**: *aggressive*

  - **My identifier**: *My IP address*
  - **Peer identfier**: *User Distinguished Name*, *vpnusers@example.com*
  - **Pre-Shared Key**: *aaabbbccc* (Use something much longer and more random!)
  - **Policy Generation**: *Unique*
  - **Proposal Checking**: *Strict*
  - **Encryption Algorithm**: *AES 128*
  - **Hash Algorithm**: *SHA1*
  - **DH Key Group**: *2*
  - **Lifetime**: *86400*
  - **NAT Traversal**: *Force*
  - Click **Save**

Phase 2 settings

- Click  inside the Mobile Phase 1 to expand its Phase 2 list.
- Click  to add a new Phase 2
- Enter the following settings:
  - **Mode**: *Tunnel*
  - **Local Network**: *(the local network, e.g. LAN, or 0.0.0.0/0 to send everything over VPN)*
  - **Protocol**: *ESP*
  - **Encryption Algorithms**: *AES 128* **only**
  - **Hash Algorithms**: *SHA1* **only**
  - **PFS key group**: *off*
  - **Lifetime**: *28800*
- Add additional phase 2 entries for additional local networks if necessary
- Click **Save**
- Click **Apply Changes**

User Settings

- Navigate to **System > User Manager**
- *Add a user*, grant the user the *User - VPN - IPsec xauth Dialin* permission, or add them to a group with this permission.
  - Note that for xauth, the password used is the password for the user, not the "IPsec Pre-Shared Key" field. That is used for non-xauth IPsec.

Firewall Rules

Don't forget to add firewall rules to pass traffic from clients

- **Firewall > Rules**, **IPsec** tab
- Add rules that match the traffic that should be allowed, or add a rule to pass any protocol/any source/any destination to allow everything.

IPsec SA Preference

- **System > Advanced**, **Miscellaneous** tab.
- Uncheck **Prefer Old IPsec SA**

34.24.2 Device Setup (Android)

Note: These settings are not present on all Android devices. See *Remote Access Mobile VPN Client Compatibility* for more info.

- Tap Settings, Networks & Wireless, VPN Settings, Advanced IPsec VPNs
- From there, press the menu button, then add.
- **Connection Template**: *PSK v1 (AES, xauth, aggressive)*
- **VPN Name**: *pfSense VPN* (Or some other description)
- **VPN Server**: *IP of the server*
  - The phone forces the keyboard to numbers, not sure if a hostname is supported.
- **Pre-Shared Key Type**: *text*
- **Pre-Shared Key**: *PSK from the Phase 1 above*
- **Identity Type**: **User FQDN**
- **Identity**: *vpnusers@example.com*
- **Username**: *xauth username*
- **Password**: *xauth password*
- **Internal Subnet IP**: *Whatever subnet(s) were specified in Phase 2 above*.
- Finish

34.24.3 Device Setup (iOS)

- Tap **Settings > General > Network > VPN**
- Tap **Add VPN Configuration**
- Tap **IPsec**
- **Description**: *pfSense VPN* (Or some other description)
- **Server**: *IP of the server*

- **Account**: *xauth username*
- **Password**: *xauth password* (or leave blank to be prompted every time)
- **Group Name**: *vpnusers@example.com*
- **Secret**: *PSK from the Phase 1 above*

34.24.4 Troubleshooting

By default iOS will tunnel all traffic over the VPN, including traffic going to the Internet. If Internet sites are inaccessible once connected, a DNS server may need to be pushed to the client for it to use, such as the LAN IP address of the firewall if the DNS forwarder is enabled, or a public DNS server such as 8.8.8.8/8.8.4.4.

The reason for the above is that the 3G provider is likely giving mobile devices DNS servers that are only accessible from their network. Once connected to the VPN the DNS servers are now being accessed via the VPN instead of the 3G network, and the queries are likely to be dropped. Supplying a local/public DNS server will work around that.

34.25 Configuring IPsec IKEv2 Remote Access VPN Clients

Most operating systems include native client support for IPsec IKEv2 VPN connections, and others typically have an app or add-on package which adds the capability.

This section covers IPsec IKEv2 client configuration for several popular operating systems.

See also:

pfSense Hangouts on Youtube to watch the September and October 2015 Hangouts on Remote Access VPNs which covers client installations for most operating systems.

34.25.1 Configuring IPsec IKEv2 Remote Access VPN Clients on Windows

Windows 8 and newer easily support IKEv2 VPNs, and Windows 7 can as well though the processes are slightly different. The procedure in this section was performed on Windows 10, but Windows 8 is nearly identical. The procedure to import certificates to Windows 7 can be found on the strongSwan Wiki

Import the CA to the Client PC

- Export the CA Certificate from pfSense® and download or copy it to the client PC:
  - Navigate to **System > Cert Manager**, **Certificate Authorities** tab on pfSense
  - Click  by the CA to download only the certificate
- Locate the downloaded file on the client PC (e.g. VPNCA.crt) as seen in Figure *Downloaded CA Certificate*
- Double click the CA file
- Click **Install Certificate...** as shown in *Certificate Properties*
- Select **Local Machine** as shown in *Certificate Import Wizard - Store Location*
- Click **Next**
- Click **Yes** at the UAC prompt if it appears

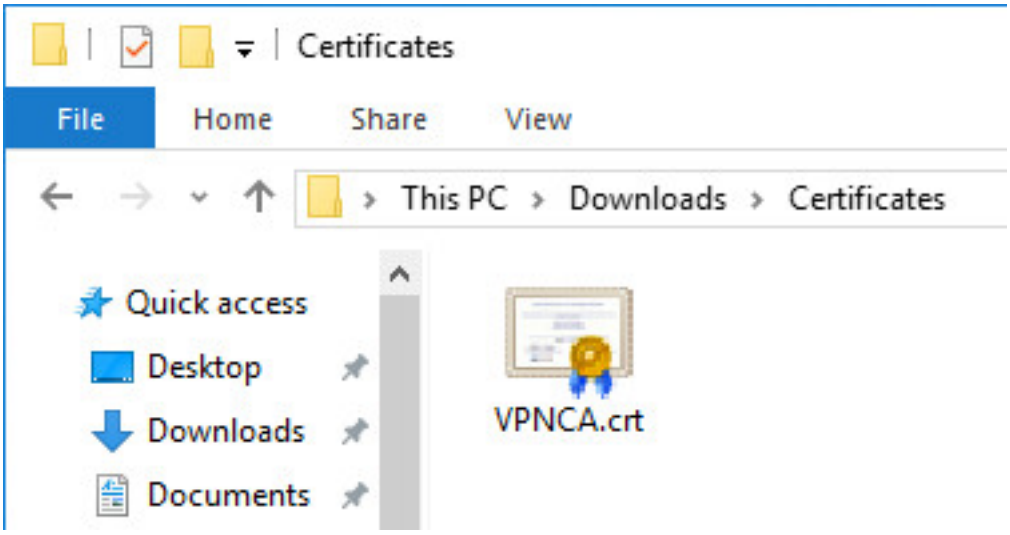


Fig. 11: Downloaded CA Certificate

- Select **Place all Certificates in the following store** as shown in Figure *Certificate Import Wizard - Browse for the Store*
- Click **Browse**
- Click **Trusted Root Certification Authorities** as shown in Figure *Select Certificate Store*
- Click **Next**
- Review the details, they should match those in Figure *Completing the Certificate Import Wizard*
- Click **Finish**
- Click **OK**
- Click **OK**

Setup the VPN Connection

Once the certificate has been properly imported it is time to create the client VPN connection. The exact steps will vary depending on the version of Windows being used by the client, but will be close to the following procedure.

- Open **Network and Sharing Center** on the client PC
- Click **Set up a new connection or network**
- Select **Connect to a workplace**
- Click **Next**
- Select **No, create a new connection**
- Click **Next**
- Click **Use my Internet Connection (VPN)**
- Enter the IP address or hostname of the server into the **Internet address** field as shown in Figure *Windows IKEv2 VPN Connection Setup Screen*

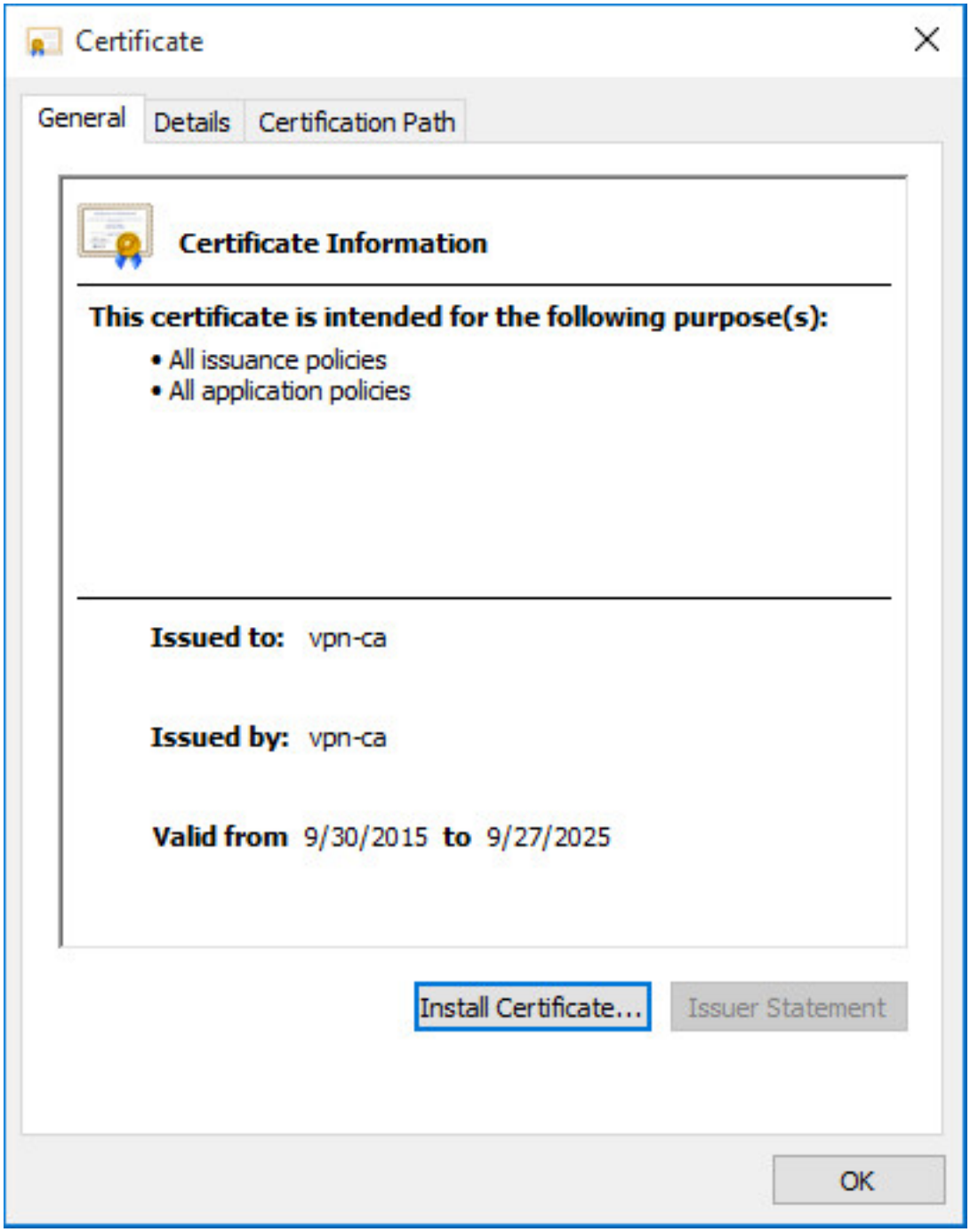


Fig. 12: Certificate Properties

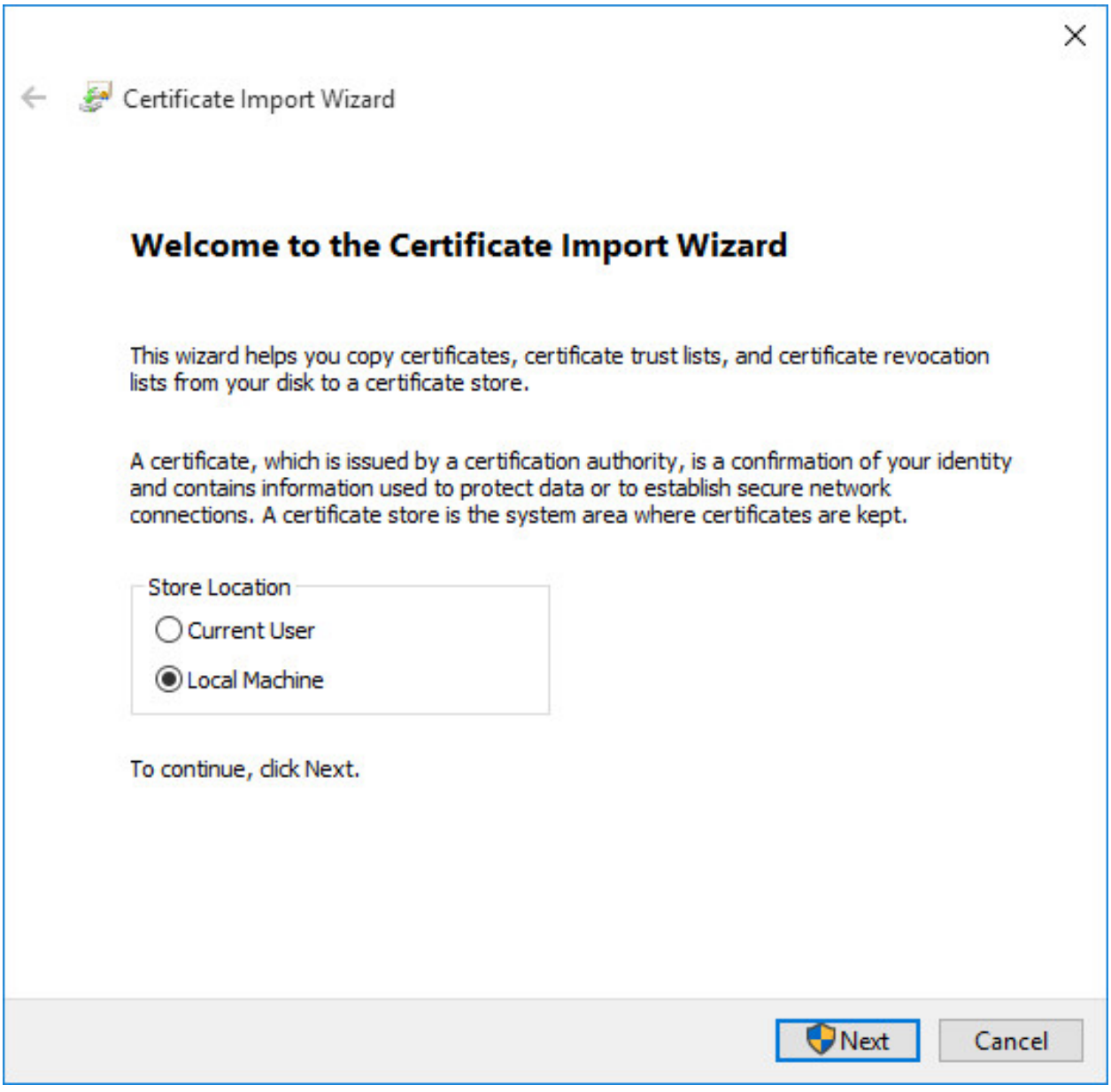


Fig. 13: Certificate Import Wizard - Store Location

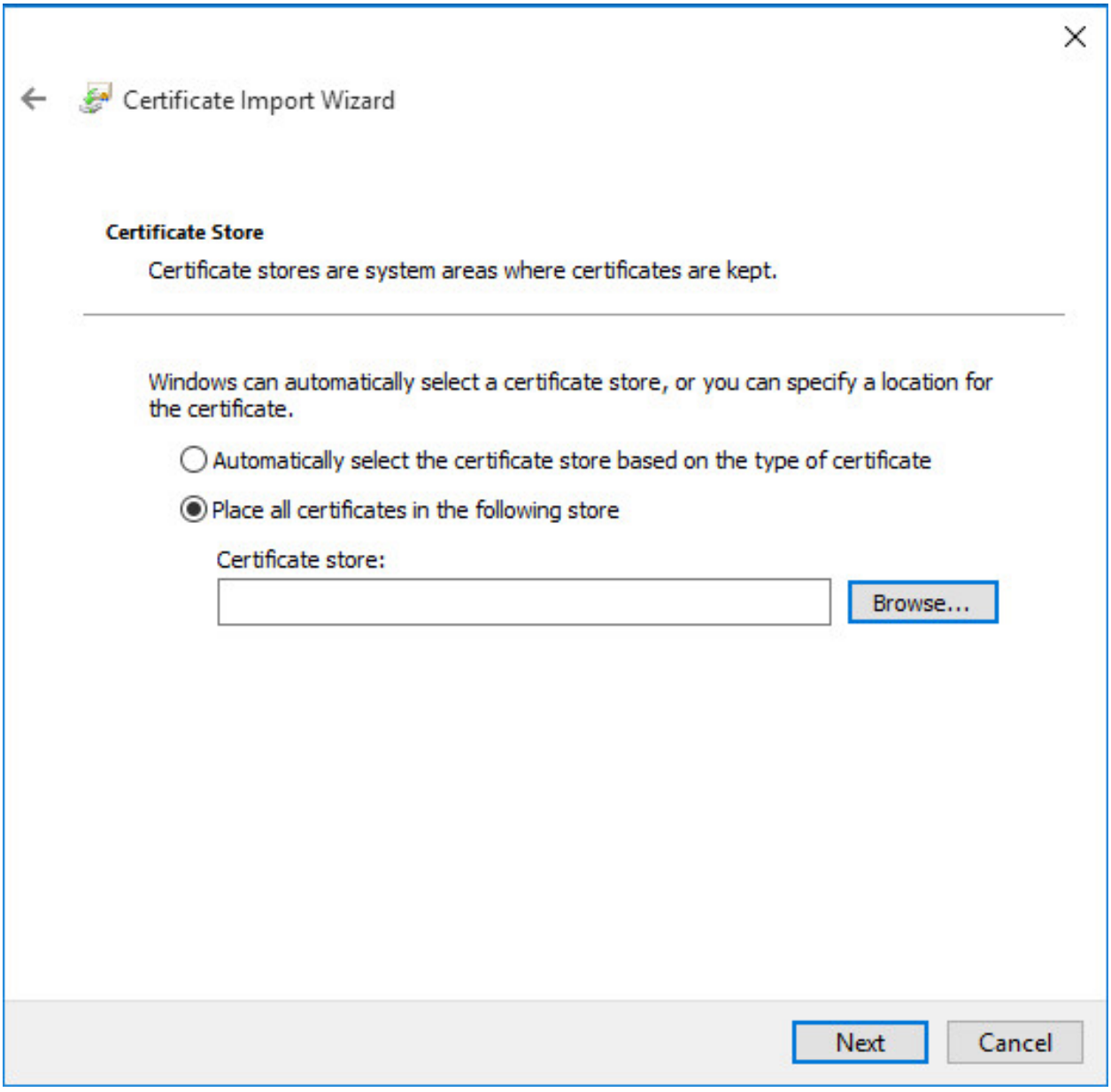


Fig. 14: Certificate Import Wizard - Browse for the Store

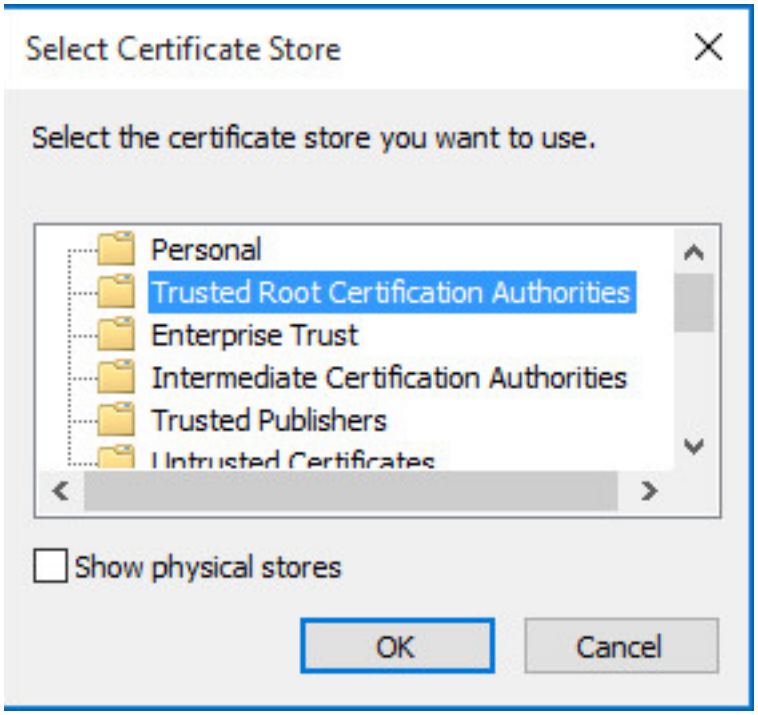


Fig. 15: Select Certificate Store

Note: This **must** match what is in the server certificate Common Name or a configured Subject Alternative Name!

- Enter a **Destination Name** to identify the connection
- Click **Create**

The connection has been added but with several undesirable defaults. For example the type defaults to *automatic*. A few settings need to be set by hand first to ensure a proper connection is made. Refer to Figure *Windows IKEv2 VPN Connection Properties*

- In **Network Connections / Adapter Settings in Windows**, find the connection created above
- Right click the connection
- Click **Properties**
- Click the **Security** tab
- Set **Type of VPN** to *IKEv2*
- Set **Data Encryption** to *Require Encryption (disconnect if server declines)*
- Set **Authentication / Use Extensible Authentication Protocol** to *Microsoft: Secured password (EAP-MSCHAP v2) (encryption enabled)*
- Compare the values on the screen to those in Figure *Windows IKEv2 VPN Connection Properties*
- Click **OK**

The connection is now ready to use.

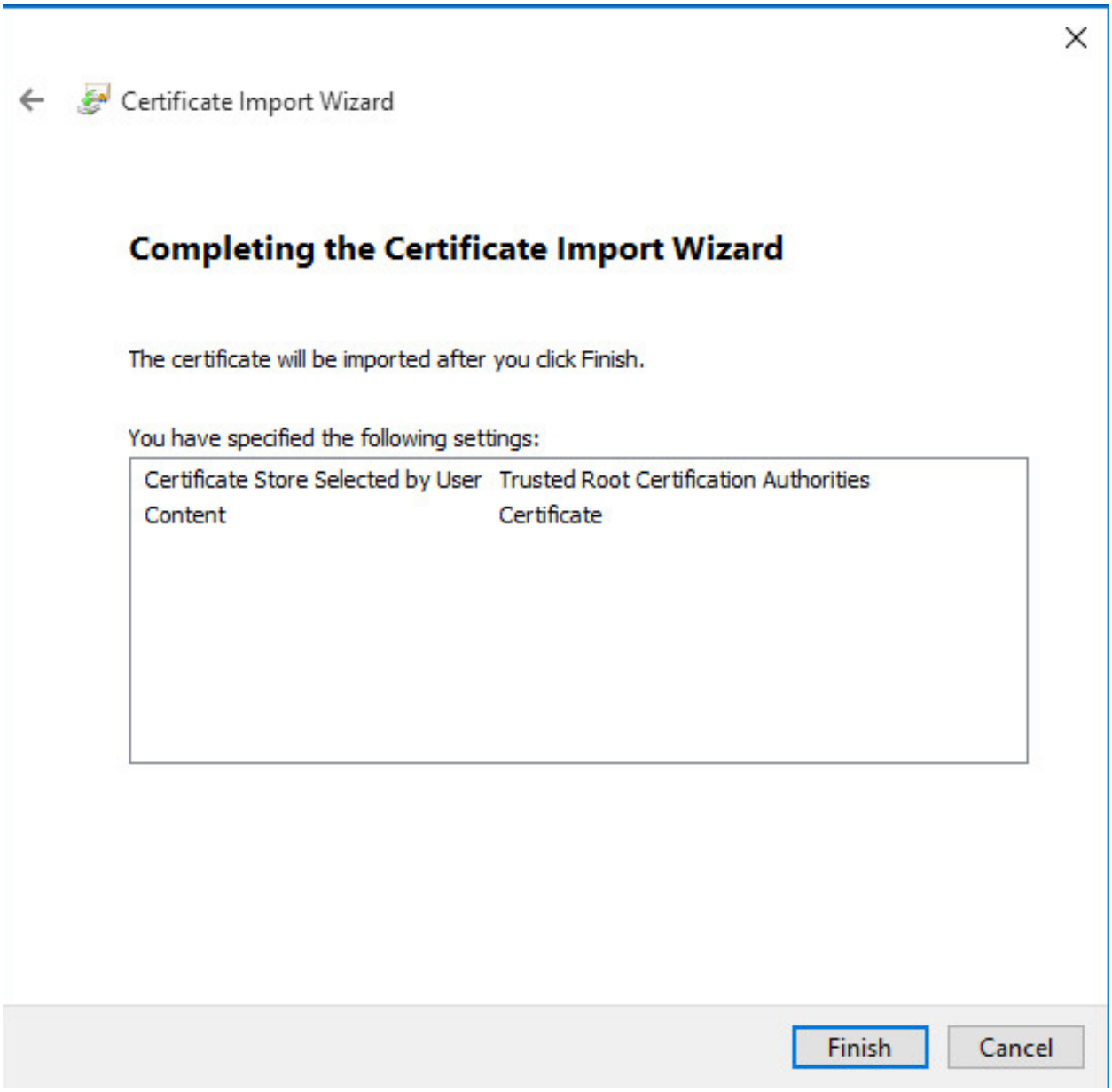


Fig. 16: Completing the Certificate Import Wizard

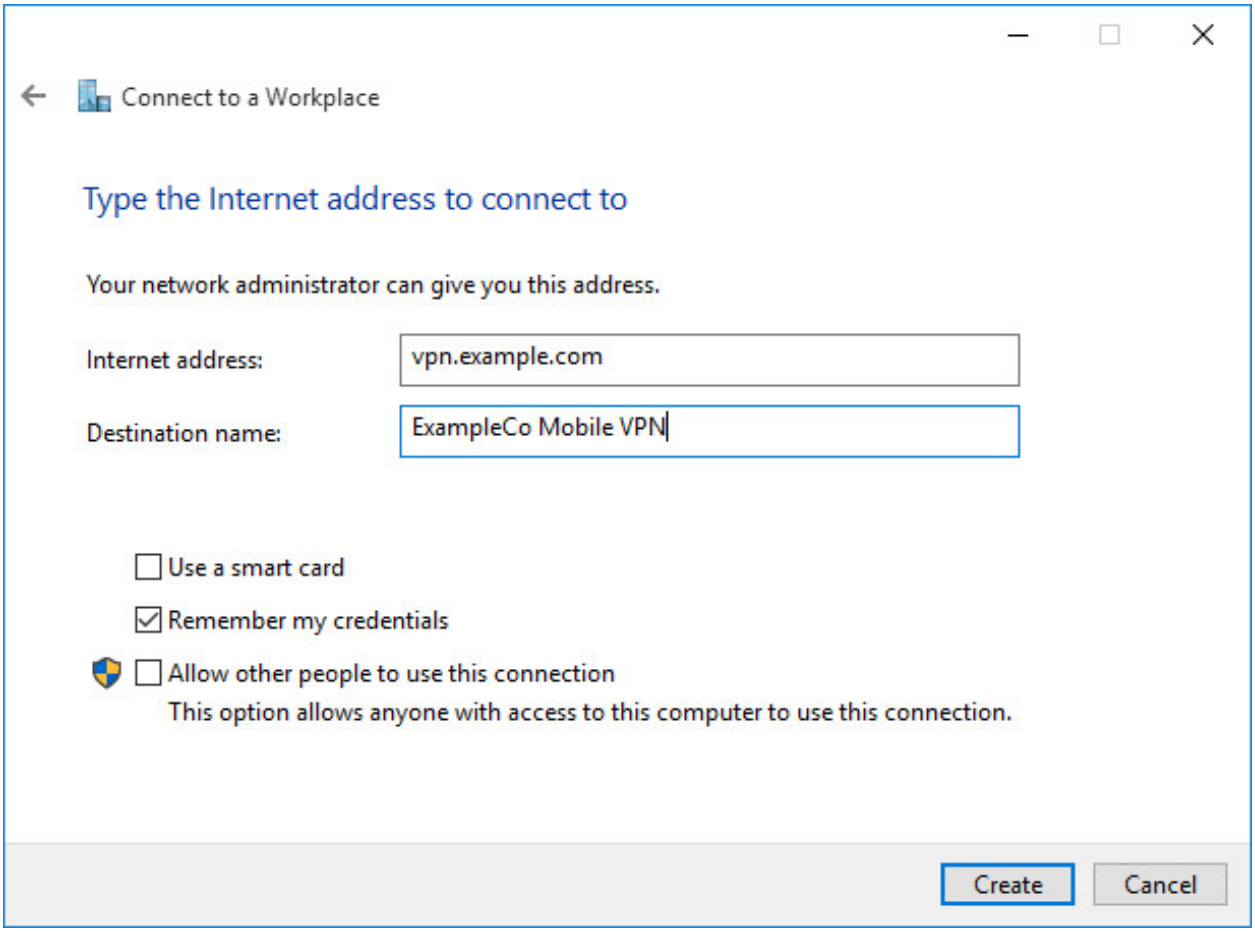


Fig. 17: Windows IKEv2 VPN Connection Setup Screen

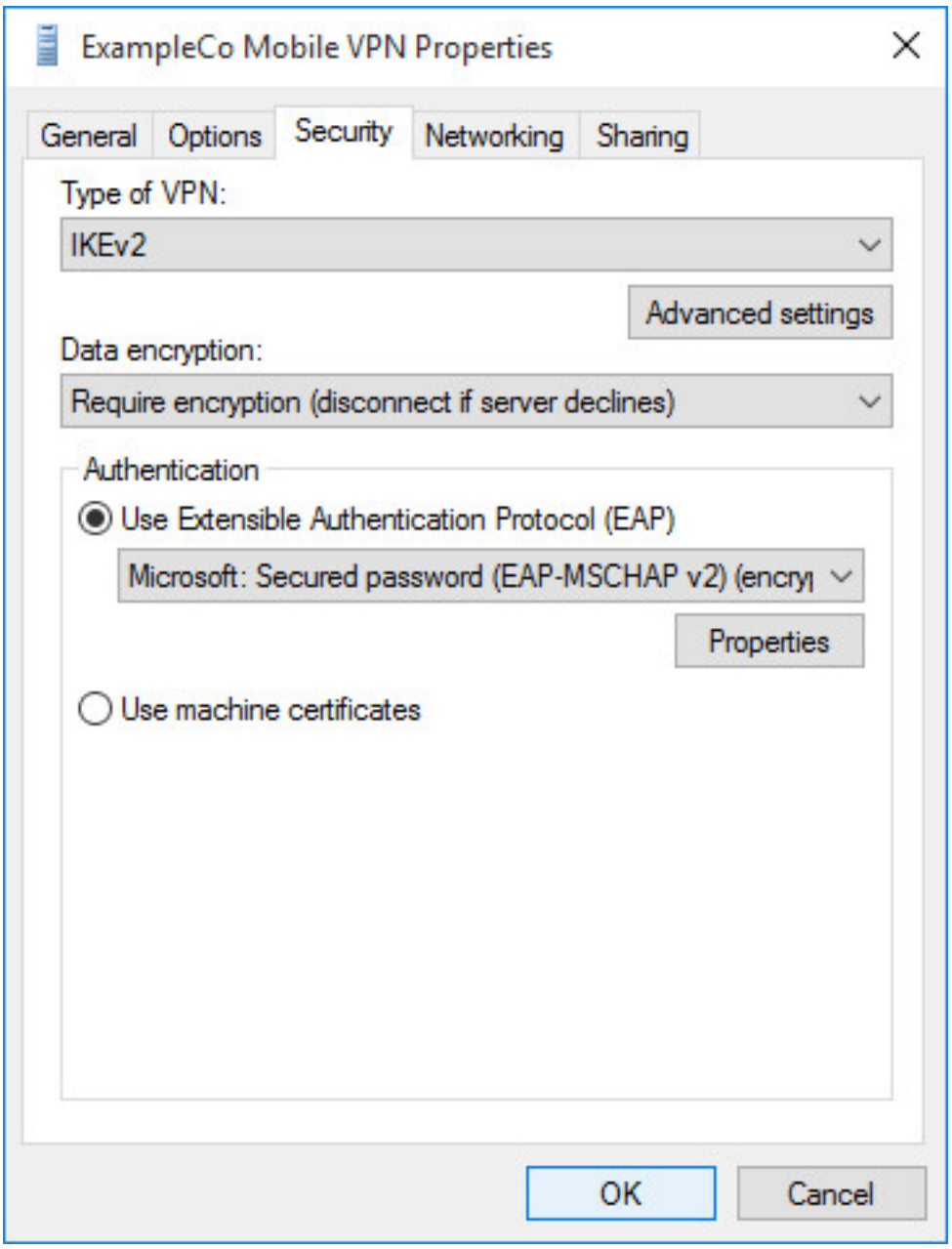


Fig. 18: Windows IKEv2 VPN Connection Properties

Disable EKU Check

When the CA and server certificates are made properly on pfSense 2.2.4 and later, this is not necessary. If an improperly generated server certificate must be used for some reason, then the Extended Key Usage check may need to be disabled on Windows. Disabling this check also disables validation of the certificate's common name and SAN fields, so it is potentially dangerous. Any certificate from the same CA could be used for the server when this is disabled, so proceed with caution.

To disable the extended key usage checks, open up **Registry Editor** on the Windows client and navigate to the following location in the client registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\RasMan\Parameters\
```

In there, add a new **DWORD** entry named **DisableIKENameEkuCheck** and set it to `1`.

A reboot may be required to activate the setting.

Advanced Windows IPsec settings

With Windows 10 PowerShell cmdlet `Set-NetIPsecMainModeCryptoSet` it is possible to change various advanced settings, like IPsec lifetime:

```
PS C:\>Set-NetIPsecMainModeCryptoSet -DisplayGroup "pfSense IPsec" -MaxMinutes 600
```

This example modifies the maximum IPsec SA lifetime for the "pfSense IPsec" connection. The default Windows IPsec lifetime is 4800 minutes (eight hours).

See also:

For more information, see Windows 10 IPsec PowerShell cmdlets

34.25.2 Configuring IPsec IKEv2 Remote Access VPN Clients on Android

Note: Android considers using a VPN an action that must be secure. When activating any VPN option the OS will force the user to add some form of locking to the device if one is not already present. It doesn't matter which type of lock is chosen (PIN lock, Pattern lock, Password, etc) but it will not allow a VPN to be configured until a secure lock has been added. On Android devices with Face lock, that is not available as a secure lock type.

Before starting, install the strongSwan app from the Play Store:

Setup the VPN Connection

- Copy the CA Certificate to the device
- Open the strongSwan app
- Import the CA:
  - Tap the settings icon (Three vertical dots in the upper right)
  - Tap **CA Certificates**
  - Tap the settings icon (Three vertical dots in the upper right)
  - Tap **Import Certificate**

  - Locate the CA Certificate copied earlier and tap it.
- Tap **Add VPN Profile**
- Enter a **Profile Name** (optional, if left blank, the gateway address will be used)
- Enter the address of the firewall as the **Gateway** (e.g. `vpn.example.com`)
- Select *IKEv2 EAP (Username/Password)* for the **Type**
- Enter the **Username**
- Enter the **Password** to have it be remembered or leave it blank to prompt for the password on each connection.
- Check Select automatically under CA Certificate
- Compare the settings to Figure *Android strongSwan Client Settings*

Connecting and Disconnecting

To Connect:

- Open the strongSwan app
- Tap the desired VPN
- Check **I trust this application** at the security prompt as shown in *Android strongSwan Client Settings*
- Tap OK

To Disconnect:

- Swipe down from the top notification bar
- Tap the strongSwan entry in the notification list
- Tap **Disconnect**

Alternately:

- Open the strongSwan app
- Tap **Disconnect** on the desired VPN

34.25.3 Configuring IPsec IKEv2 Remote Access VPN Clients on OS X

As of OS X 10.11 (El Capitan) it is possible to configure an IKEv2 type VPN manually in the GUI without needing a VPN Profile configuration file. Configuration for IKEv2 is integrated into the network management settings the same as other connections. Before a client can connect, however, the VPN Server's CA Certificate must be imported.

Import the CA Certificate into OS X

- Copy the CA Certificate to the OS X system
- Double click the CA Certificate File in Finder (Figure *OS X Certificate File in Finder*), which opens Keychain Access
- Locate the imported certificate under **Login**, **Certificates** as shown in Figure *OS X Keychain Access Login Certificate List*
- Drag the certificate on to **System**
- Enter the login credentials and click **Modify Keychain**

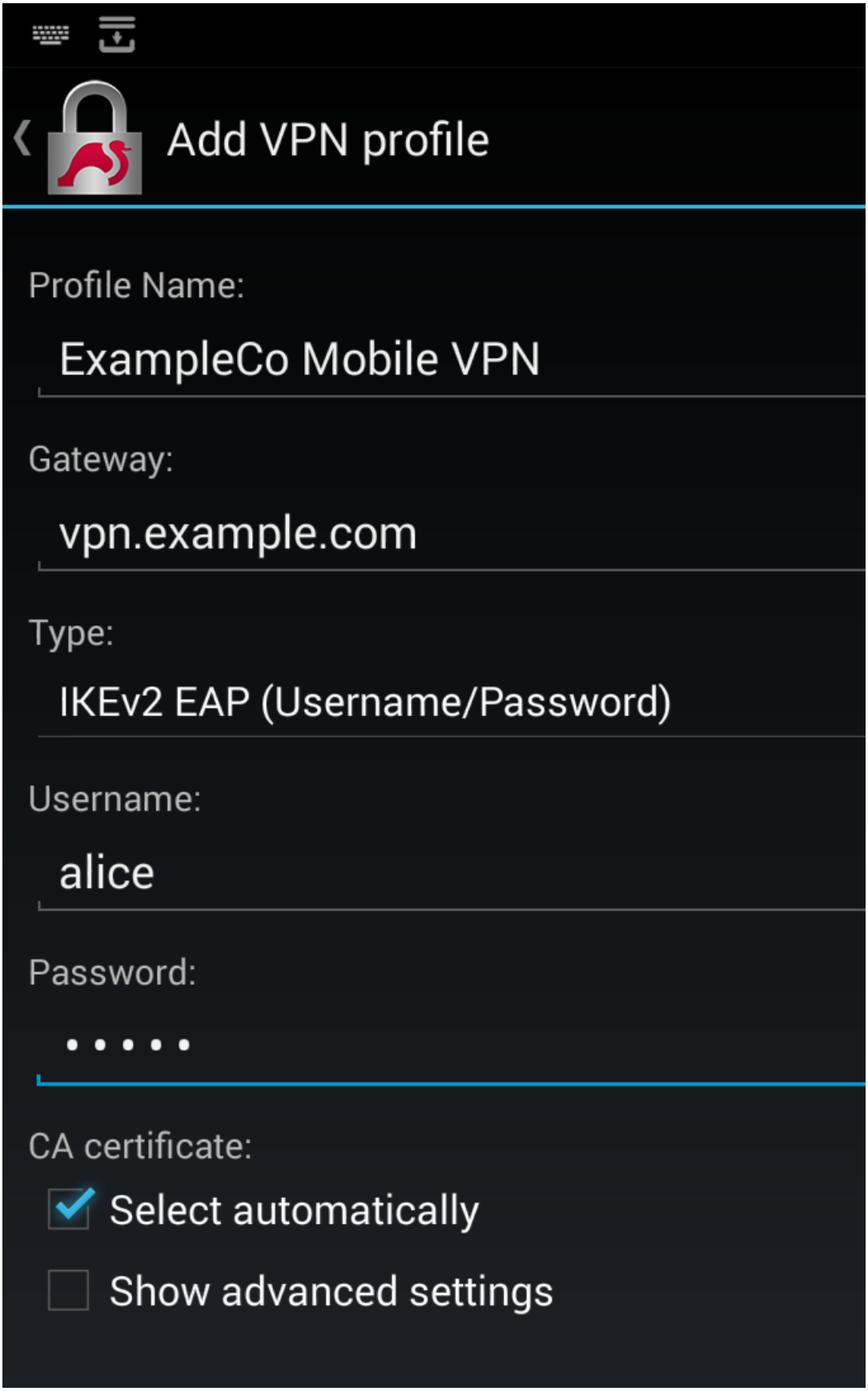


Fig. 19: Android strongSwan Client Settings

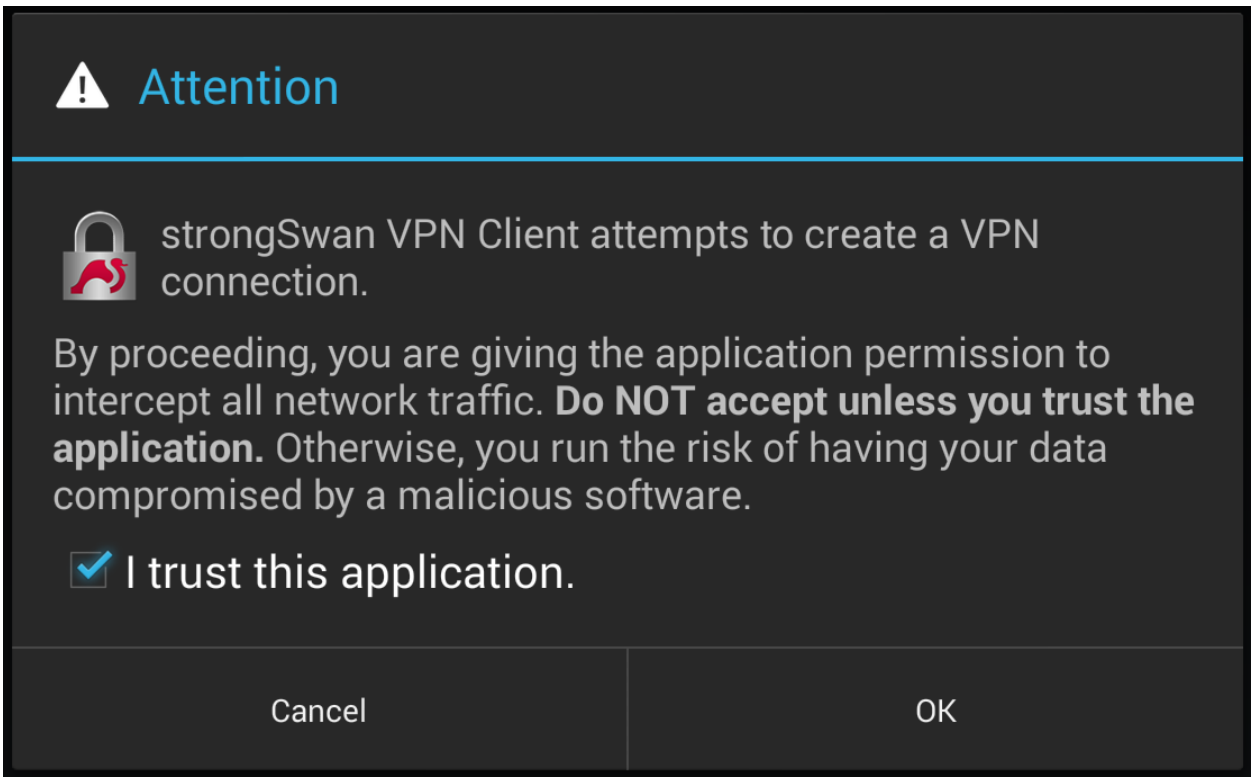


Fig. 20: Android strongSwan Client Settings

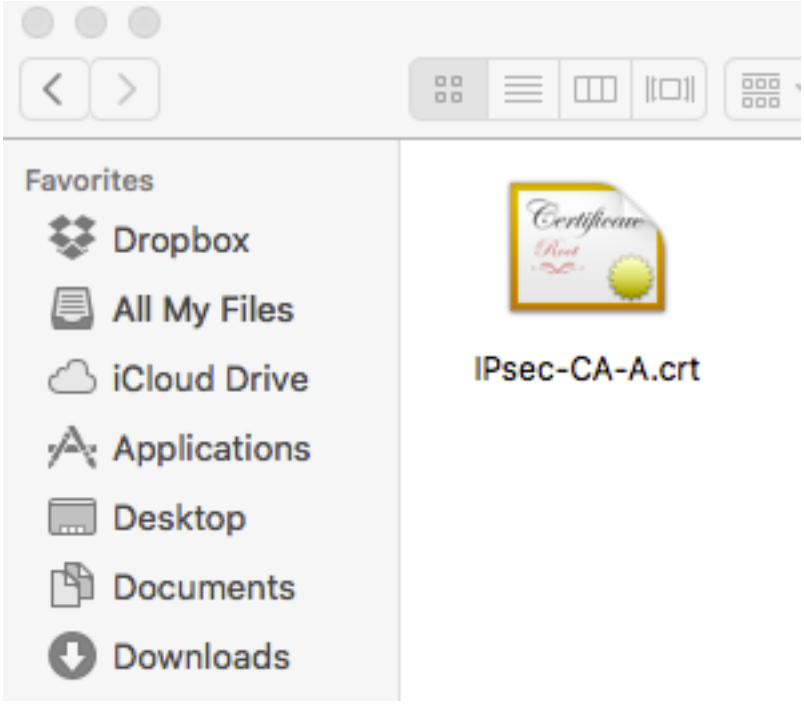


Fig. 21: OS X Certificate File in Finder

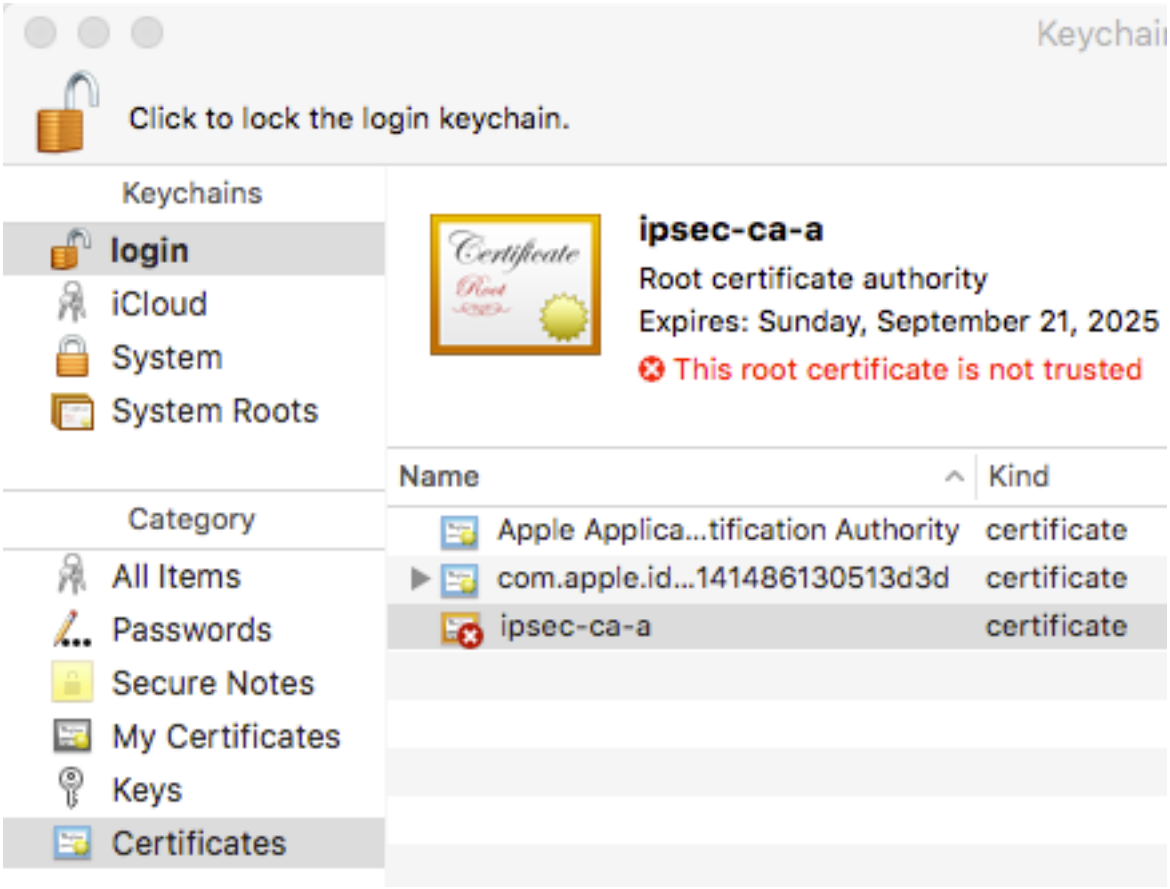


Fig. 22: OS X Keychain Access Login Certificate List

- Locate the imported certificate under **System**, **Certificates** as shown in Figure *OS X Keychain Access System Certificate List*

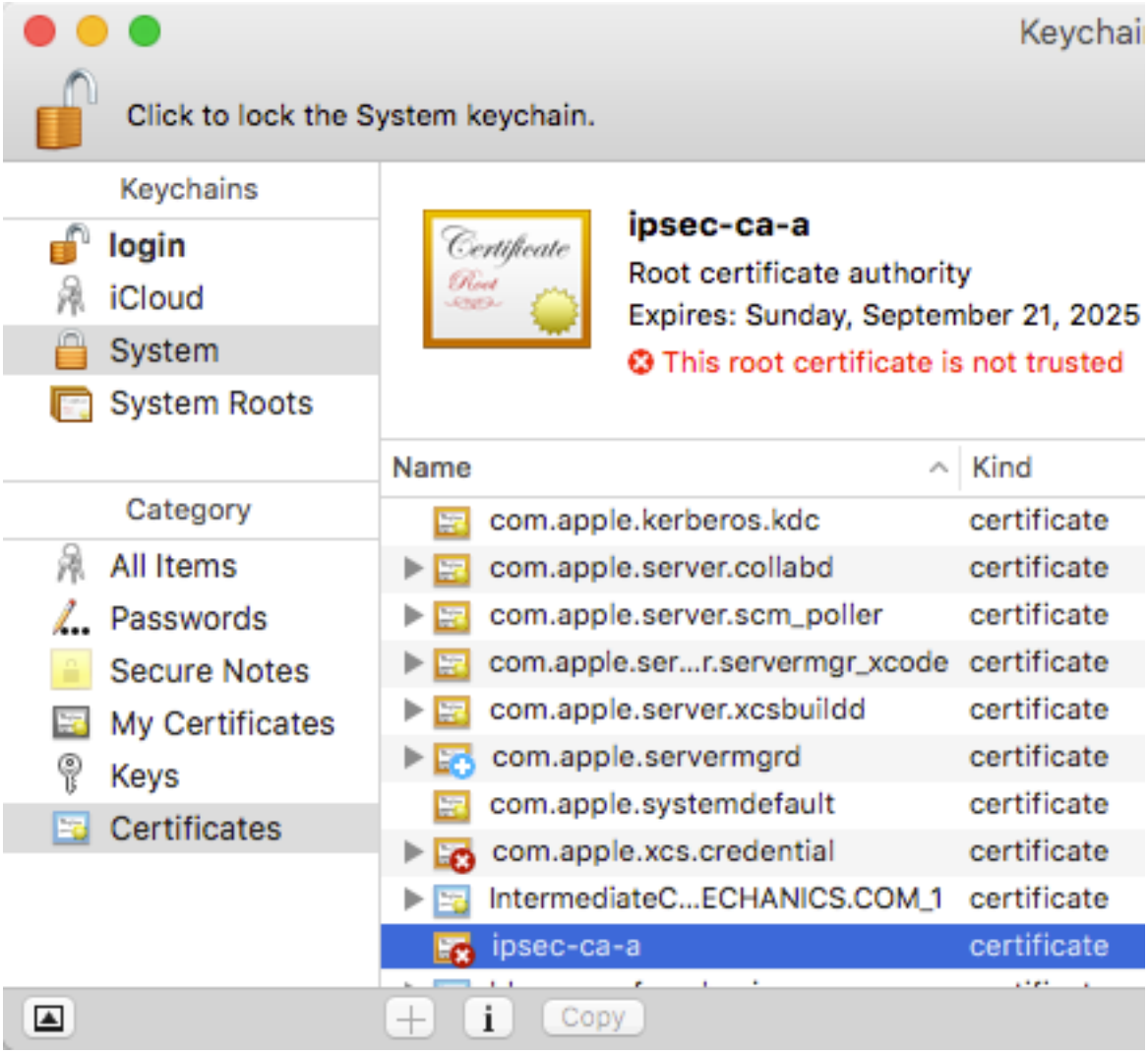


Fig. 23: OS X Keychain Access System Certificate List

- Click the Certificate
- Click **File > Get Info**
- Expand **Trust**
- Set **When using this certificate** to *Always Trust* as shown in Figure *OS X Certificate Trust Settings*
- Click the red close button to close the certificate info window, which will cause an authentication prompt to allow the change.
- Enter the login credentials and click **Update Settings**
- Quit Keychain Access

The certificate is now located in System Certificates and has been marked as trusted so it can be used for the VPN.



Fig. 24: OS X Certificate Trust Settings

Setup the VPN Connection

- Open System Preferences
- Click **Network**
- Click the lock icon and enter credentials to make changes if the settings have not already been unlocked
- Click + to add a new VPN entry as shown in Figure *OS X Add Network Button*

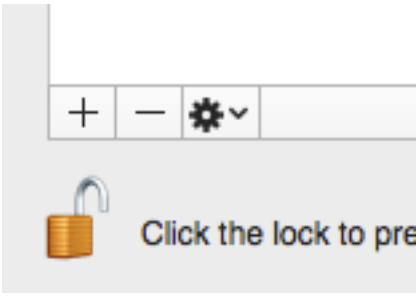


Fig. 25: OS X Add Network Button

- Select *VPN* for the **Interface**
- Select *IKEv2* for the **VPN Type** (default)
- Set **Service Name** to a description for the VPN (e.g. `ExampleCo VPN`) to complete the form, which will look similar to Figure *OS X Create VPN Prompt*
- Click **Create**
- Enter the hostname of the firewall in DNS as the **Server Address**
- Enter the hostname of the firewall again in **Remote ID**

Note: This must match the server certificate's Common Name and SAN entry.

- Leave **Local ID** blank, the settings will now look like Figure *OS X IKEv2 VPN Settings*
- Click **Authentication Settings**
- Select **Username**

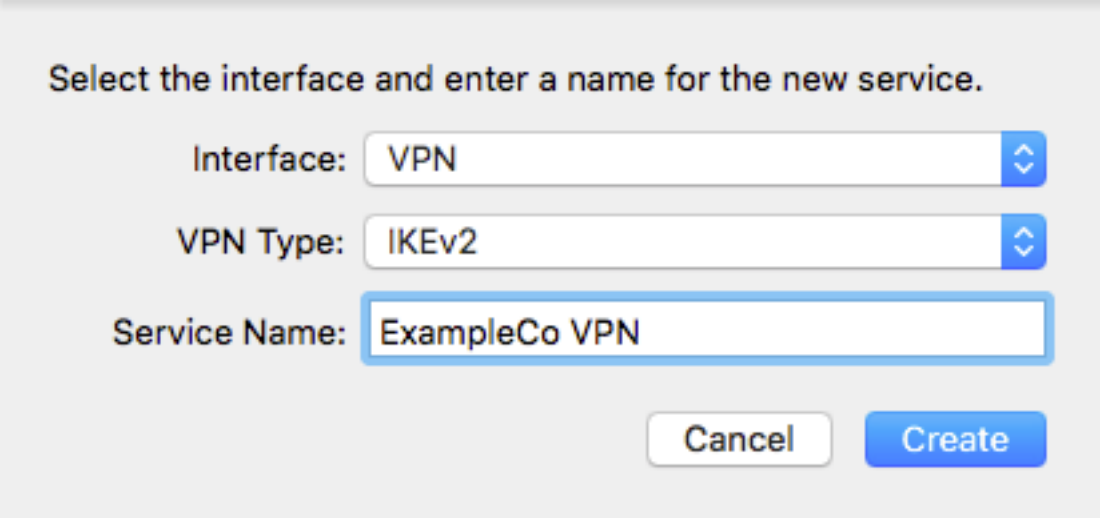


Fig. 26: OS X Create VPN Prompt

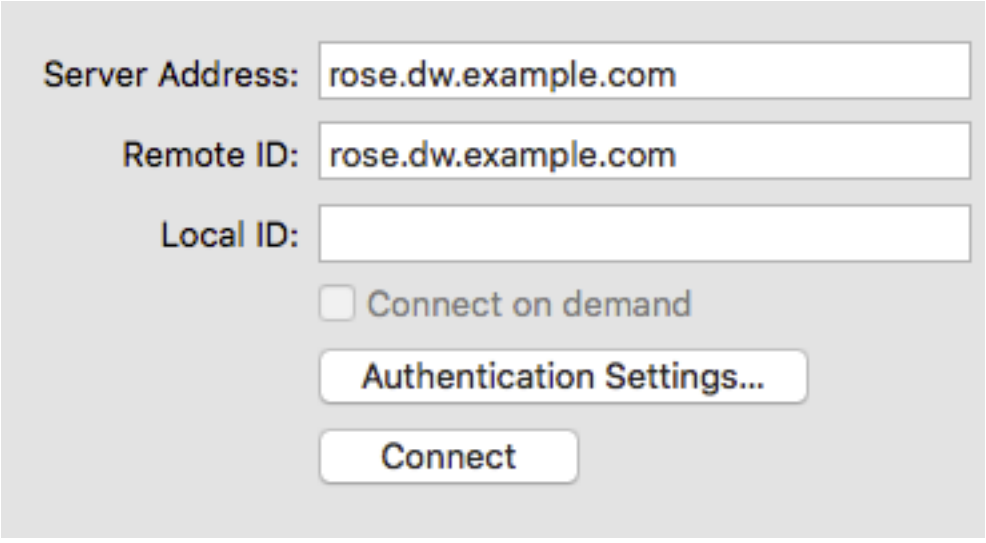


Fig. 27: OS X IKEv2 VPN Settings

- Enter the **Username** and **Password** as shown in Figure *OS X IKEv2 VPN Authentication Settings*

Note: With EAP-MSCHAPv2 the **Username** is the **Identifier** configured for the user's entry on the **Pre-Shared Keys** tab under **VPN > IPsec**. With EAP-RADIUS this would be the username set on the RADIUS server.

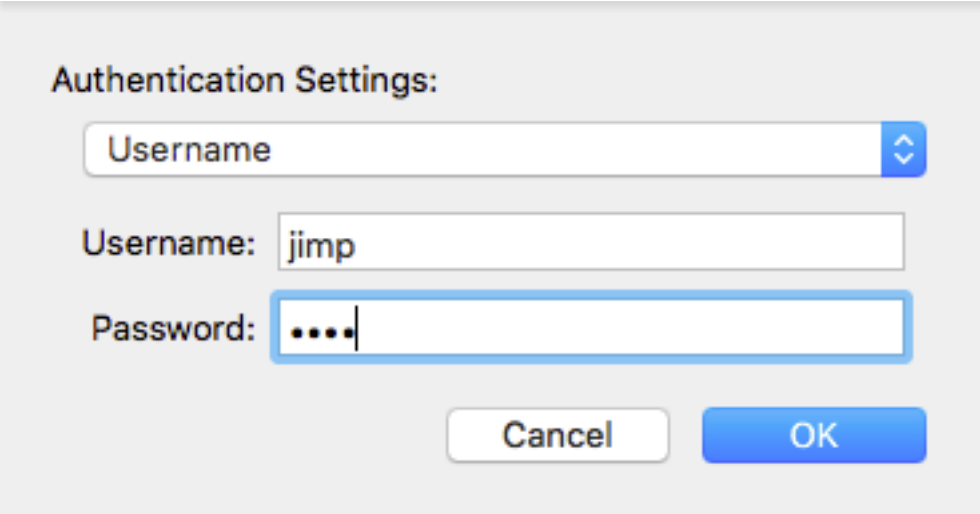


Fig. 28: OS X IKEv2 VPN Authentication Settings

- Check **Show VPN status in the menu bar** (if desired)
- Click **Apply**

Connecting and Disconnecting

Managing the connection can be done multiple ways. The first method is to click **Connect** or **Disconnect** on the VPN entry in Network settings. The second, easier method is to check **Show VPN Status in the menu bar** in the VPN settings and then manage the connection from that icon, as shown in Figure *OS X VPN Status Menu*.

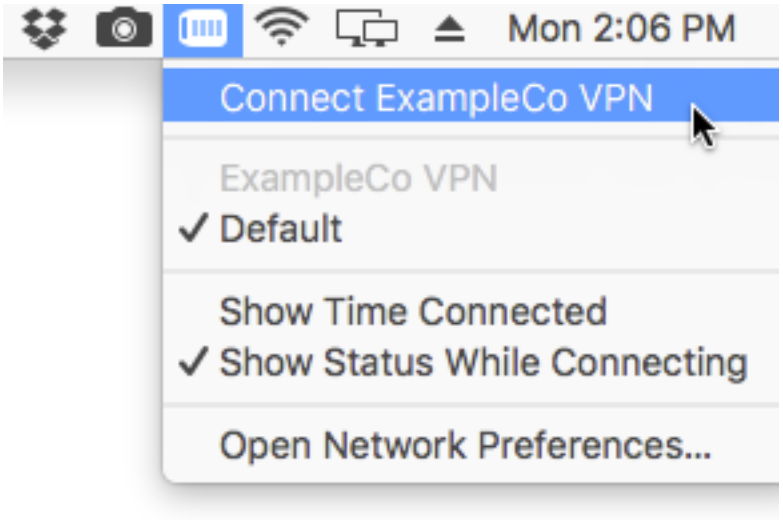


Fig. 29: OS X VPN Status Menu

34.25.4 Configuring IPsec IKEv2 Remote Access VPN Clients on iOS

As of version 9, iOS has built-in support for IKEv2 that can be configured from the GUI without requiring a VPN Profile. As with other clients, the CA Certificate must be installed.

Import the CA to the iOS Device

Importing the CA Certificate to the client device is a relatively easy process. The first step is to get the CA Certificate to the client device. The easiest way to accomplish this is via e-mail as shown in Figure *iOS Mail Client Receiving CA Certificate*

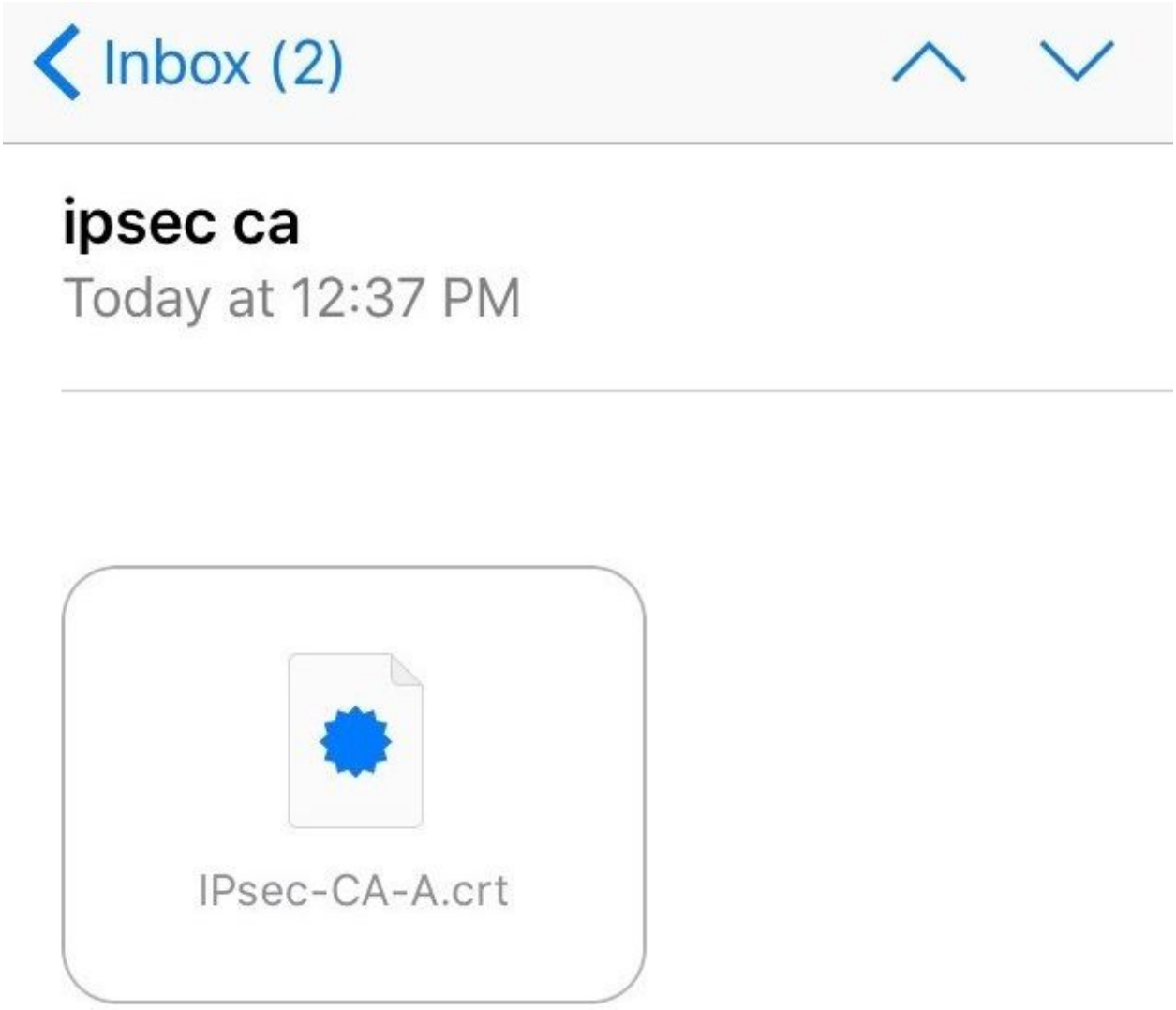


Fig. 30: iOS Mail Client Receiving CA Certificate

To install the certificate from e-mail:

- Send the CA Certificate only (not the key) to an e-mail address reachable from the client device
- Open the Mail app on the client device

- Open the message containing the CA Certificate
- Tap the attachment to install the CA Certificate and the **Install Profile** prompt will show as seen in *iOS CA Certificate Install Profile Prompt*

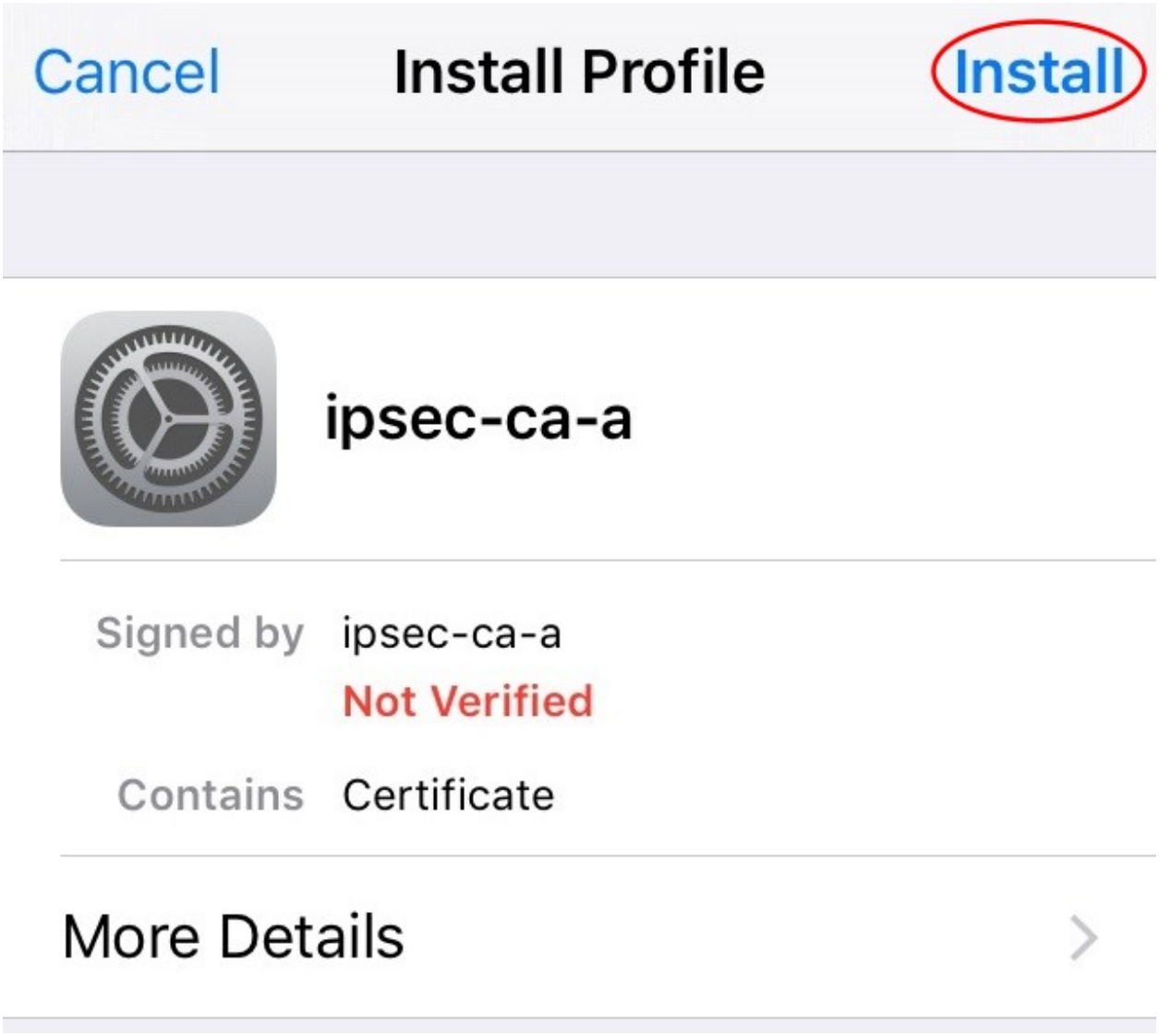


Fig. 31: iOS CA Certificate Install Profile Prompt

- Tap **Install** in the upper right, and a warning screen is presented as shown in *iOS CA Certificate Install Warning*
- Tap **Install** in the upper right once more to confirm and then one final prompt is presented as seen in *iOS CA Certificate Confirmation Prompt*
- Tap **Install** at the confirmation prompt and the CA Certificate is now stored as a trusted entry.

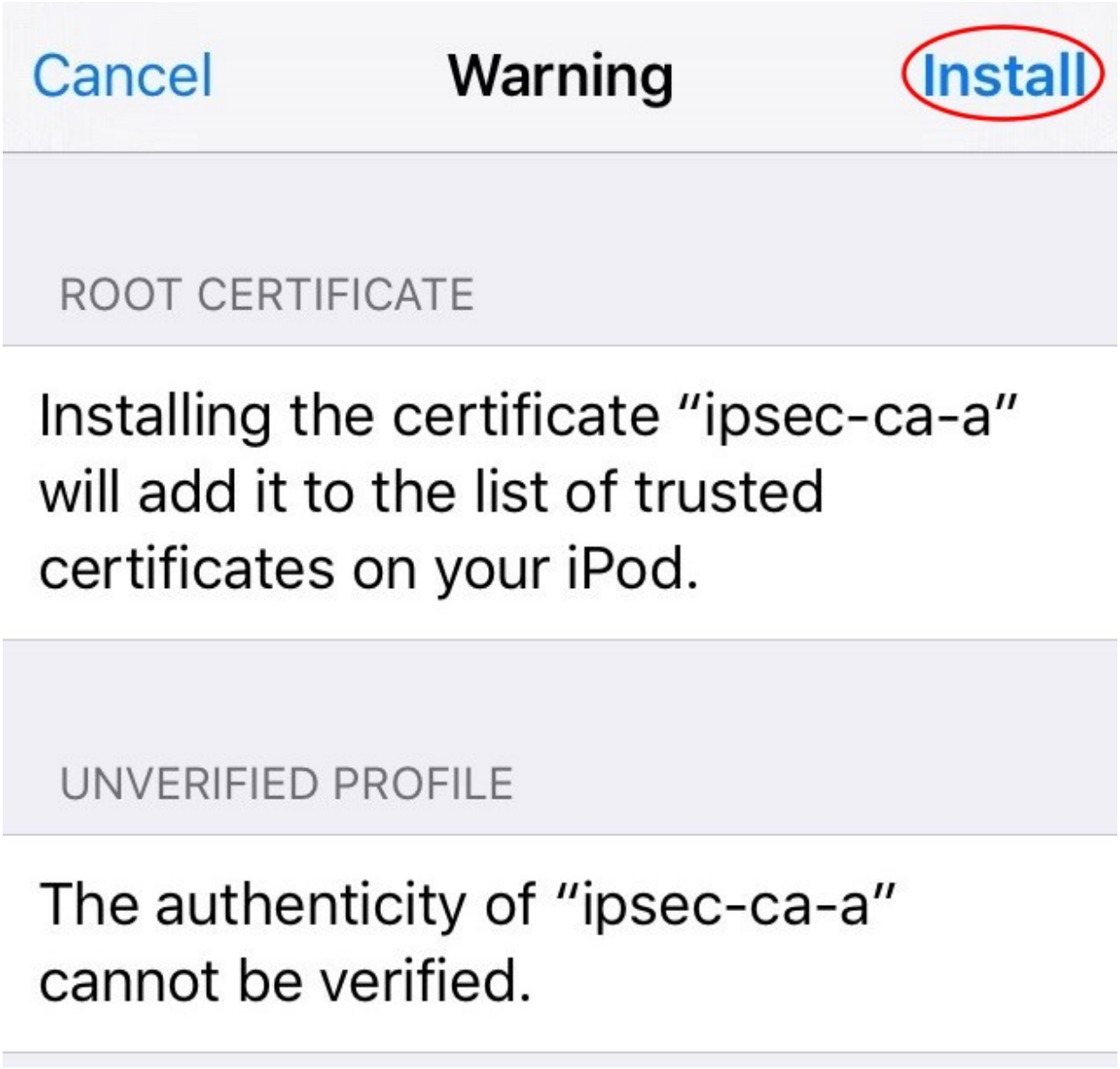


Fig. 32: iOS CA Certificate Install Warning

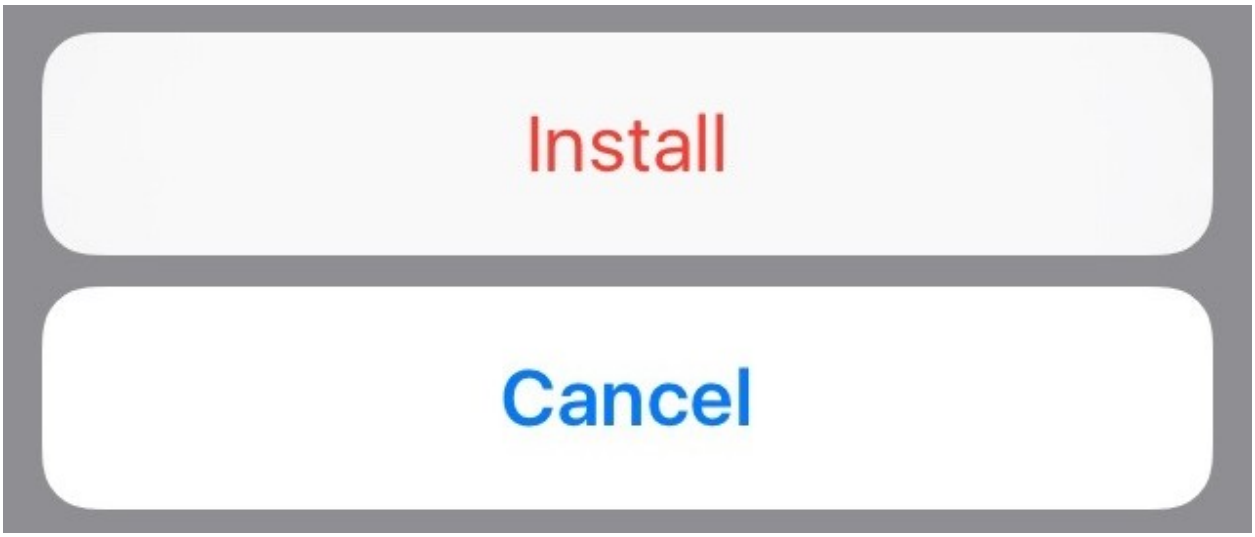


Fig. 33: iOS CA Certificate Confirmation Prompt

Setup the VPN Connection

Once the CA Certificate has been installed, a VPN entry must be configured:

- Open **Settings**
- Tap **General**
- Tap **VPN**
- Tap **Add VPN Configuration**
- Set **Type** to *IKEv2* (default)
- Enter some text for the **Description** (e.g. **ExampleCo VPN**)
- Enter the hostname of the firewall in DNS as the **Server**
- Enter the hostname of the firewall again in **Remote ID**

Note: This must match the server certificate's Common Name and SAN entry.

- Leave **Local ID** blank
- Set **User Authentication** to *Username*
- Enter the **Username** and **Password**

Note: With EAP-MSCHAPv2 the **Username** is the **Identifier** configured for the user's entry on the **Pre-Shared Keys** tab under **VPN > IPsec**. With EAP-RADIUS this would be the username set on the RADIUS server.

- Tap **Done** to complete the VPN entry. When complete, it looks similar to *iOS IKEv2 Client Settings*

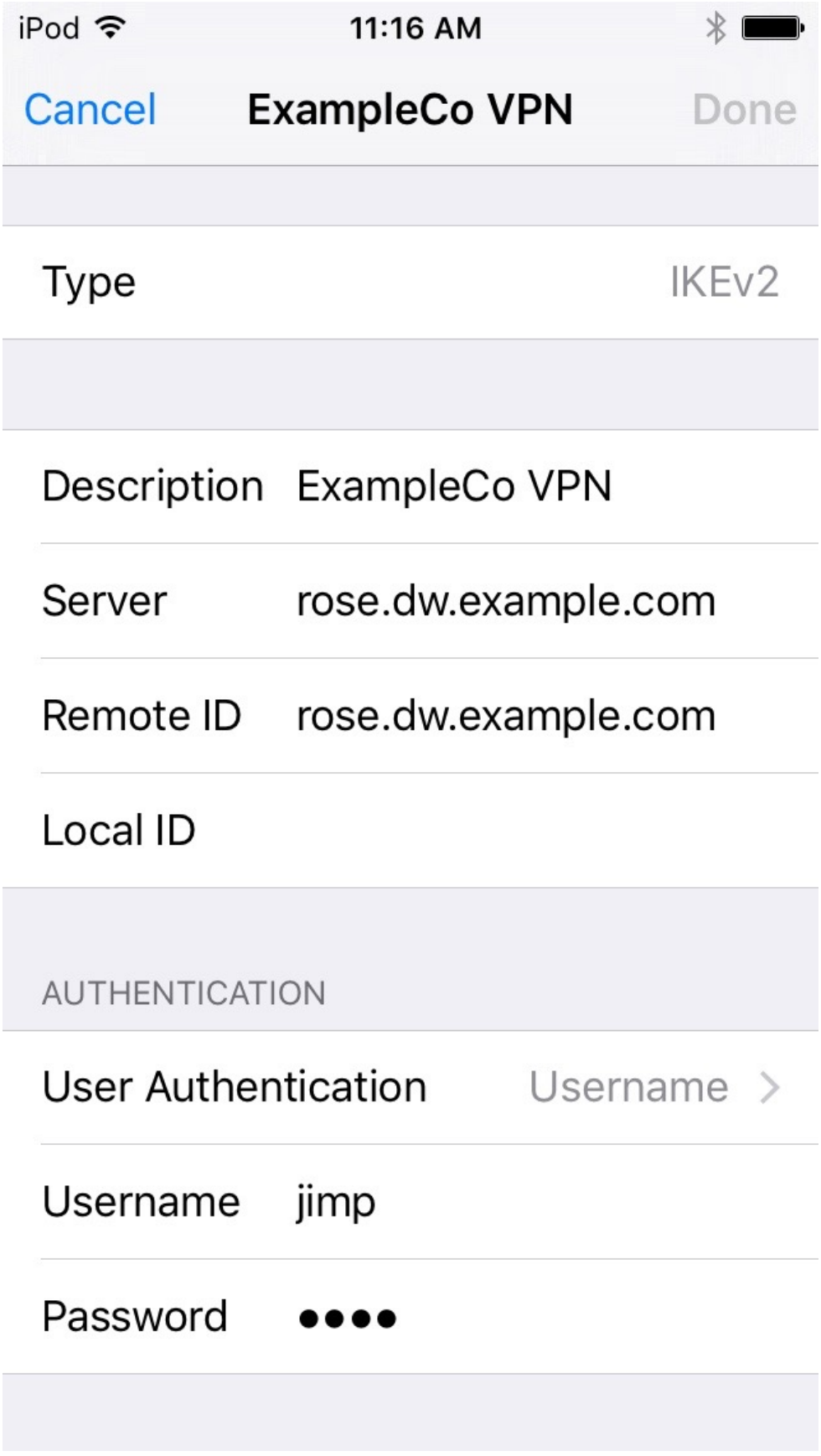


Fig. 34: iOS IKEv2 Client Settings

Connecting and Disconnecting

The VPN may be connected or disconnected by visiting the VPN entries under **Settings**. This varies a bit but typically shows in at least two places:

1. Settings > VPN
2. Settings > General > VPN

The entry directly under **Settings** appears near the top of the list with the other Network entries (Airplane mode, Wi-Fi, and Bluetooth) once there is at least one VPN connection present.

Once in the VPN list, the VPN entry must be selected (shows a checkmark next to its entry) and then the slider may be moved to the "On" position to connect.



Fig. 35: iOS VPN List

34.25.5 Configuring IPsec IKEv2 Remote Access VPN Clients on Ubuntu

Before starting, install **network-manager-strongswan** and **strongswan-plugin-eap-mschapv2** using `apt-get` or a similar mechanism.

Setup the VPN Connection

- Copy the CA Certificate for the VPN from the firewall to the workstation
- Click the **Network Manager** icon in the notification tray by the clock (Icon varies depending on the type of network in use)
- Click **Network Connections**
- Click **Add**
- Select *IPsec/IKEv2 (strongswan)* under **VPN** as shown in *Adding an IKEv2 VPN on Ubuntu*

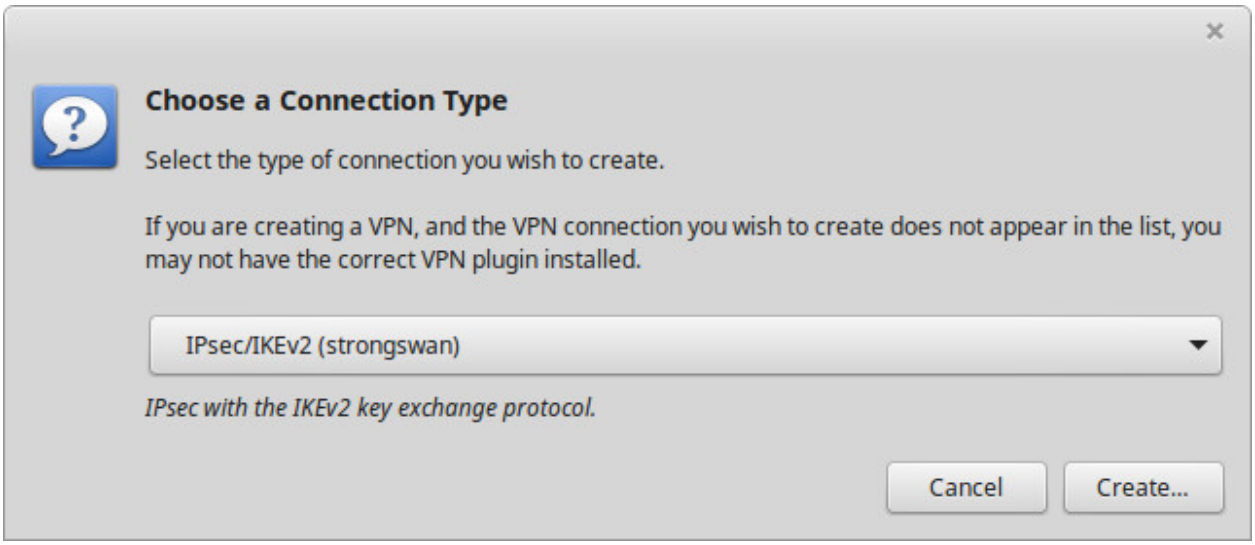


Fig. 36: Adding an IKEv2 VPN on Ubuntu

Note: If the option is not present, double check that **network-manager-strongswan** is installed.

- Click **Create**
- Enter a **Description** (e.g. `ExampleCo Mobile VPN`)
- Select the **VPN** Tab
- Enter the **Address** of the firewall (e.g. `vpn.example.com`)
- Select the control next to **Certificate** and browse to find the downloaded CA Certificate
- Select *EAP* for **Authentication**
- Enter the **Username** to be used for this connection (e.g. `alice`)
- Check **Request an inner IP address**
- Compare the settings to those shown in figure *Ubuntu VPN Client Settings*

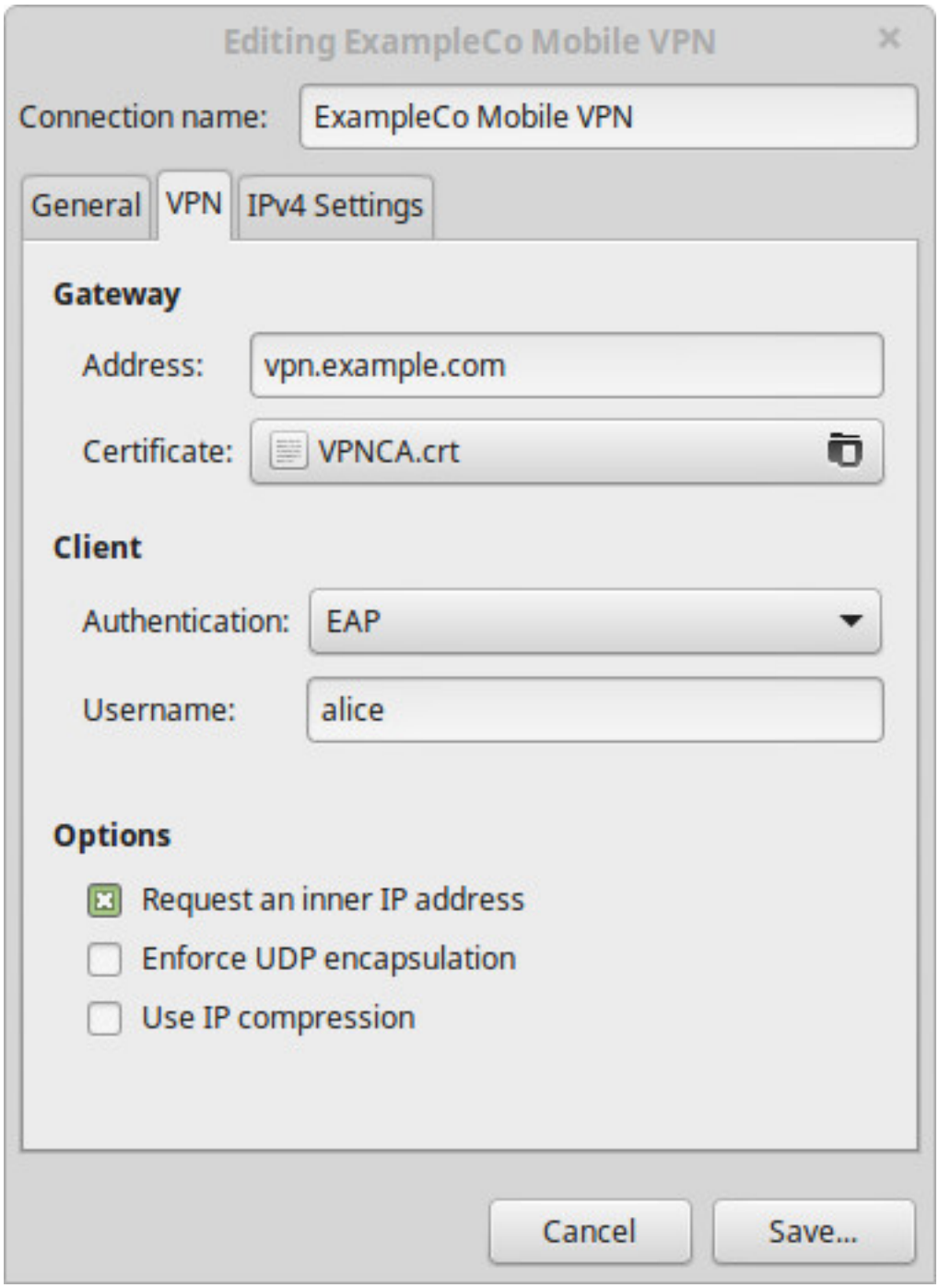


Fig. 37: Ubuntu VPN Client Settings

- Click **Save**
- Click **Close**

Connecting and Disconnecting

To Connect:

- Click the **Network Manager** icon
- Click the VPN Name or click **VPN Connections** to move the slider to the *On (1)* position

---

Note: If a password prompt does not appear, the network manager service may need restarted or a reboot of the workstation may be necessary.

To Disconnect:

- Click the **Network Manager** icon
- Click **VPN Connections** to move the slider to the *Off (0)* position

34.26 IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2

34.27 IKEv2 Server Configuration

There are several components to the server configuration for mobile clients:

- Creating a certificate structure for the VPN
- Configuring the IPsec **Mobile Client** settings
- Creating the phase 1 and phase 2 for the client connection
- Adding IPsec firewall rules.
- Create user credentials for the VPN

34.27.1 IKEv2 Certificate Structure

Create a Certificate Authority

If a suitable Certificate Authority (CA) is not present in the Cert Manager, creating one is the first task:

- Navigate to **System > Cert Manager** on the pfSense® firewall
- Click  **Add** to create a new certificate authority
- Select *Create an internal Certificate Authority* for the **Method**
- Fill in the rest of the fields as desired with company or site-specific information
- Click **Save**

Create a Server Certificate

Warning: Follow these directions exactly, paying close attention to how the server certificate is created at each step. If any one part is incorrect, some or all clients may fail to connect.

- Navigate to **System > Cert Manager**, **Certificates** tab on the pfSense firewall
- Click  **Add** to create a new certificate
- Select *Create an internal certificate* for the **Method**
- Enter a **Descriptive Name** such as `IKEv2 Server`
- Select the appropriate **Certificate Authority** created in the previous step
- Choose the desired **Key Type**, **Key length**, **Digest algorithm**, and **Lifetime**
- Enter the **Common Name** as the hostname of the firewall *as it exists in DNS*. If clients will connect by IP address, place the IP address here instead
- Fill in the regional and company values in the **Distinguished name** fields as desired, they are copied from the CA and may be left as-is
- Set the **Certificate Type** to *Server Certificate*
- Click  **Add** to add a new **Alternative Name**
- Enter `FQDN or Hostname` in the **Type** field
- Enter the hostname of the firewall *as it exists in DNS* **again** in the **Value** field
- Click  **Add** to add another new **Alternative Name**
- Enter `IP Address` in the **Type** field
- Enter the WAN IP address of the firewall in the **Value** field
- Add more **Alternative Names** as needed for additional hostnames or IP addresses on the firewall that clients may use to connect
- Click **Save**

Tip: As an alternative, the ACME package (*ACME package*) can generate a server certificate which will be trusted natively by many clients.

34.27.2 Mobile Client Settings

Before configuring a mobile IPsec instance, first choose an IP address range to use for mobile clients. Ensure that IP addresses do not overlap any existing network; The IP addresses must differ from those in use at the site hosting the mobile tunnel as well as the LAN from which the client will be connecting. In this example, *10.3.200.0/24* will be used, but it can be any unused subnet.

First, enable IPsec on the firewall if it has not already been enabled:

- Navigate to **VPN > IPsec**
- Check **Enable IPsec**
- Click **Save**

Mobile client support must also be enabled:

- Navigate to **VPN > IPsec**
- Click on the Mobile clients tab (Figure *Enable Mobile IPsec Clients*).
- Check **Enable IPsec Mobile Client Support**

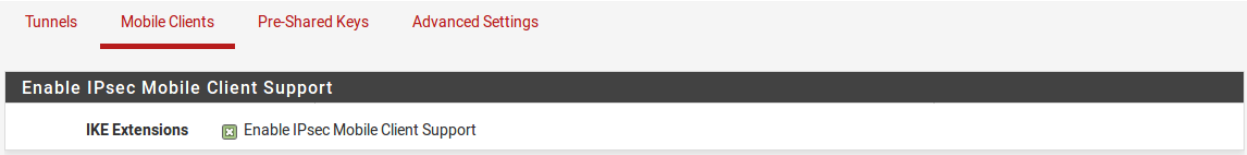


Fig. 38: Enable Mobile IPsec Clients

- Leave the authentication sources set to *Local Database*, as seen in Figure *Mobile Clients Authentication*. This setting is not needed for EAP- MSCHAPv2, but it must have something selected. RADIUS servers defined in the User Manager (*User Management and Authentication*) can be selected here for authenticating users when using EAP-RADIUS.

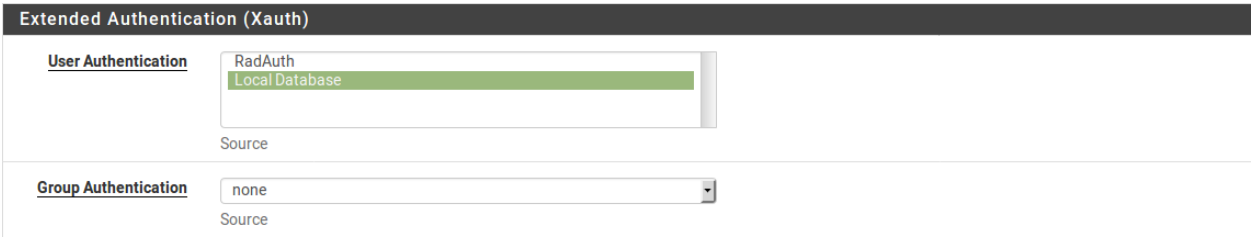


Fig. 39: Mobile Clients Authentication

Some settings may be pushed to the client, such as the client IP address and DNS servers. These options are shown in Figure *Mobile Clients Pushed Settings*. Support for these options varies between clients, but is common and well-supported in most current operating systems.

Virtual Address Pool Defines the pool of IP addresses that will be handed out to clients. Use `10.3.200.0/24` for this example.

Virtual IPv6 Address Pool Same as above, but for IPv6 addresses.

Network List Controls whether the client will attempt to send all of its traffic across the tunnel, or only traffic for specific networks. If this option is checked, then the networks defined in the **Local Network** options for the mobile phase 2 definitions will be sent to the client. If this option is unchecked, the clients will attempt to send all of their traffic, including Internet traffic, across the tunnel. Not all

clients respect this option. For this example, the client can only reach the network in the phase 2, so check this option.

Save Xauth Password When checked, clients that support this control will allow the user to save their credentials when using Xauth. This is mainly respected by Cisco-based clients like the one found on iOS and Mac OS X. Since IKEv2 is being used in this example, it is not important.

DNS Default Domain When checked, the value entered into the box will be pushed to clients as their default domain suffix for DNS requests. For example if this is set to `example.com` and a client requests `host`, then the DNS request will be attempted for `host.example.com`.

Split DNS Controls how the client will send DNS requests to the DNS Server supplied (if any). If this option is unchecked, the client will send all of its DNS requests to a provided DNS Server. If the option is checked, but left empty, and a **DNS Default Domain** is set, then only requests for that domain name will go to the provided DNS Server. If it's checked and a value is entered, then only requests for the domain(s) entered in the box will be forwarded to the provided DNS Server. In this example, both `example.com` and `example.org` are used and DNS requests for those two domains will to go to the VPN servers, so enter those values here separated by a space.

DNS Servers When **Provide a DNS server list to clients** is checked, and IP addresses are entered for the local DNS servers, such as `10.3.0.1`, these values are sent to clients for use while the VPN is connected.

Note: If mobile clients will route to the Internet over the VPN, ensure the clients get a DNS Server from the firewall using this option, and that they do not have **Split DNS** enabled. If this is not done, the clients will attempt to get DNS from whatever server they were assigned by their ISP, but route the request across the tunnel and it will most likely fail.

WINS Servers Works similar to DNS servers, but for WINS. Rarely used these days, best left disabled.

Phase 2 PFS Group Overrides the PFS setting for all Mobile Phase 2 entries. Generally best to set this value on the P2 entries individually, so leave unchecked.

Login Banner Optional, and only works on Xauth clients. Leave unchecked and blank.

- Click **Save** and pfSense will display a warning that there is no phase 1 definition for mobile clients
- Click **Create Phase 1** to make a new Phase 1 entry for mobile clients
- Click the **Tunnels** tab

Phase 1

The Phase 1 configuration for mobile clients is presented, and must be configured as follows:

Key Exchange Version Set to *V2*

Internet Protocol Set to *IPv4* for this example

Interface Set to *WAN*

Description Set to `Mobile IPsec`

Authentication Method Set to *EAP-MSCHAPv2*

My identifier Choose *Distinguished Name* from the drop-down list and then enter the hostname of the firewall, same as it was entered into the server certificate, `vpn.example.com`

Peer Identifier Set to *Any*

My Certificate Choose the IPsec Server Certificate created earlier

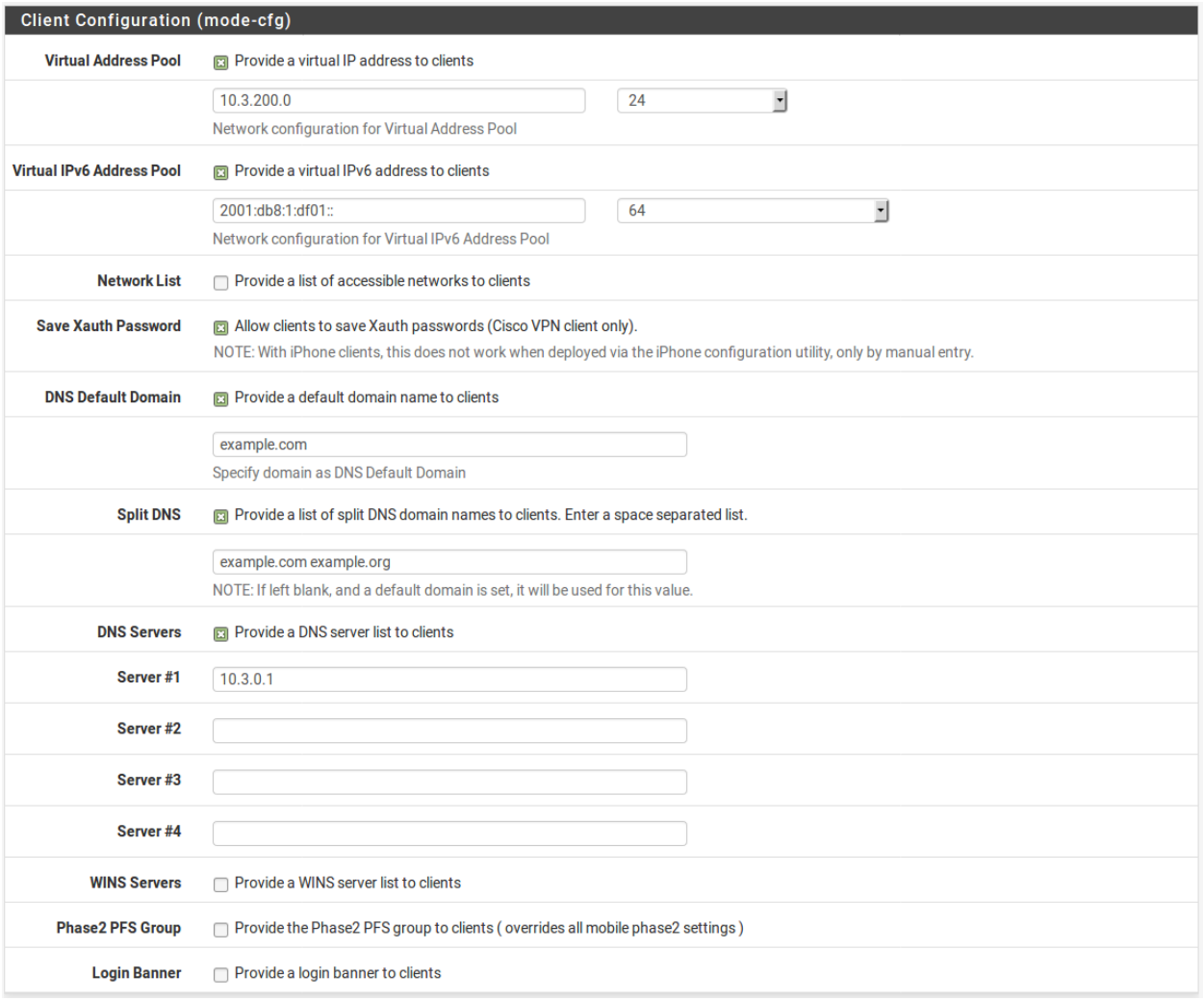


Fig. 40: Mobile Clients Pushed Settings



Fig. 41: Mobile Clients Phase 1 Creation Prompt

My Certificate Authority Choose the Certificate Authority created earlier

Encryption Algorithm

Algorithm Set to *AES*

Key Length Set to *256 bits*

Hash Set to *SHA256*

DH Group Set to *14 (2048 bit)*

Multiple combinations of encryption, hash, and DH options may be created to accommodate various clients with different requirements. Click  **Add Algorithm** to add more entries.

Lifetime Must be set to `28800`

Disable Rekey Leave unchecked

Disable Reauth Leave unchecked

Responder Only Leave unchecked

MOBIKE Set to *Enable* to allow clients to roam between IP addresses, otherwise set to *Disable*.

DPD

Enable DPD *Checked*

Delay `10` seconds

Max failures `5`

- Click **Save**

Phase 2

- Click  **Show Phase 2 Entries** to expand the list of mobile phase 2 entries
- Click  **Add P2** to add a new mobile phase 2.

Mode Set to *Tunnel IPv4*

Local Network Set to *LAN subnet* or another local network.

To tunnel all traffic over the VPN, use *Network* and enter `0.0.0.0` with a mask of *0*

NAT/BINAT Set to *None*

Protocol Set to *ESP*, which will encrypt tunneled traffic

Encryption algorithms Set to *AES* with *Auto* selected for key length.

Hash algorithms Select *SHA256*

PFS Set to *off*

Lifetime Set to `3600`

- Click **Save**
- Click Apply changes

The tunnel setup for mobile clients is complete.

34.27.3 Mobile IPsec User Creation

The next step is to add users for use by EAP-MSCHAPv2.

- Navigate to **VPN > IPsec**, **Pre-Shared Keys** tab
- Click  **Add** to add a new key
- Configure the options as follows:

  Identifier The username for the client, can be expressed in multiple ways, such as an e-mail address like `jimp@example.com`

  Secret Type Set to *EAP* for EAP-MSCHAPv2 users

  Pre-Shared Key The password for the client, for example `abc123`
- Click **Save**
- Repeat as many times as needed for additional VPN users.

A complete user is shown in Figure *Mobile IPsec User*.

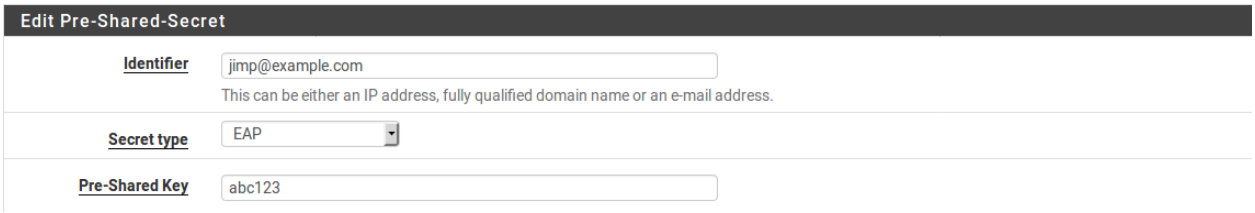


Fig. 42: Mobile IPsec User

34.27.4 Firewall Rules

As with the static site-to-site tunnels, mobile tunnels will also need firewall rules added to the **IPsec** tab under **Firewall > Rules**. In this instance the source of the traffic would be the subnet chosen for the mobile clients and the destination will be the LAN network, or *any* if tunneling all traffic. For more details, *IPsec and firewall rules*.

34.28 Client Configuration

Each mobile client computer will need to have a VPN instance added. In some cases a third-party IPsec client may be required. There are many different IPsec clients available for use, some free, and some commercial applications. With IKEv2, as used in this example, many operating systems have native VPN clients and do not need extra software.

34.29 IPsec Remote Access VPN Example Using IKEv2 with EAP-RADIUS

To setup IKEv2 with EAP-RADIUS, follow the directions for *IKEv2 with EAP-MSCHAPv2* with a slight variation:

- Define a RADIUS server under **System > User Manager**, **Servers** tab before starting
- Select the RADIUS server on **VPN > IPsec**, **Mobile Clients** tab
- Select *EAP-RADIUS* for the **Authentication method** on the Mobile IPsec Phase 1 entry

Note: When using Windows 7 as a client, be sure to import the CA Certificate from the VPN server as a machine certificate under "Computer Account" as described here.

34.29.1 EAP-RADIUS with FreeRADIUS

The default settings are OK for this, if not, see *Using EAP and PEAP with FreeRADIUS*

34.29.2 EAP-RADIUS with Windows Network Policy Server (NPS)

To allow strongSwan to authenticate against NPS using EAP-MSCHAPv2, alter the NPS policy as follows:

- Open **Network Policy Server (NPS)**
- Expand **Policies**
- Click **Network Policies**
- Edit the policy currently in use
- Click on the **Constraints** tab
- Click **Authentication Methods**
- Click **Add**
- Select **Microsoft: Secured Password (EAP-MSCHAP v2)**
- Click **OK**
- Click **Apply** (To restart NPS)
- Click **OK**

34.30 IPsec Remote Access VPN Example Using IKEv2 with EAP-TLS

Under construction. Needs testing.

IKEv2 is supported starting with pfSense® software version 2.2 and one way to make it work is by using EAP-TLS, which is covered in this article.

34.30.1 Setup Certificates

Similar to OpenVPN, a set of certificates is required for the server and clients.

Create a Certificate Authority

If one is not already available, then the first task is to create a Certificate Authority.

- Navigate to **System > Cert Manager** in the pfSense webGUI
- Click  to create a new certificate authority
- Select *Create an internal Certificate Authority* for the **Method**
- Fill in the rest of the fields as desired with company or site-specific information
- Click **Save**

Create a Server Certificate

- Navigate to **System > Cert Manager**, **Certificates** tab in the pfSense webGUI
- Click  to create a new certificate
- Select *Create an internal certificate* for the **Method**
- Enter a **Descriptive Name** such as *IKEv2 Server*
- Select the appropriate **Certificate Authority** created in the previous step
- Choose the desired **Key length**, **Digest algorithm**, and **Lifetime**
- Set the **Certificate Type** to *Server Certificate*
- Fill in the regional and company values in the **Distinguished name** fields as desired, they are copied from the CA and may be left as-is
- Enter the **Common Name** as the hostname of the firewall as it exists in DNS
- Click  to add a new **Alternative Name**
- Enter *DNS* in the **Type** field
- Enter the hostname of the firewall as it exists in DNS **again** in the **Value** field – Some clients require the value in SAN not just CN!
- Click  to add a new **Alternative Name**
- Enter *IP* in the **Type** field
- Enter the WAN IP address of the firewall in the **Value** field
- Add more **Alternative Names** as needed for additional hostnames or IP address on the firewall that clients may use to connect
- Click **Save**

Create Client Certificates

- Navigate to **System > Cert Manager**, **Certificates** tab in the pfSense webGUI
- Click  to create a new certificate
- Select *Create an internal certificate* for the **Method**
- Enter a **Descriptive Name** such as *client1*
- Select the appropriate **Certificate Authority**
- Choose the desired **Key length**, **Digest algorithm**, and **Lifetime**
- Set the **Certificate Type** to *User Certificate*
- Fill in the regional and company values in the **Distinguished name** fields as desired, they are copied from the CA and may be left as-is
- Enter the **Common Name** as the client username, such as *client1*
- Click  to add a new **Alternative Name**
- Enter *DNS* in the **Type** field
- Enter the user name again in the **Value** field, such as *client1*
- Click **Save**

Repeat as needed for additional clients.

34.30.2 Set up Mobile IPsec for IKEv2+EAP-TLS

With the certificate structure prepared, the next task is to configure the necessary IPsec settings. The settings below have been tested and found to work, but other similar settings may function as well. Feel free to try other encryption algorithms, hashes, etc. Report any additional combinations found to work or not work on the forum.

Mobile Clients

- Navigate to **VPN > IPsec**, **Mobile Clients** tab in the pfSense webGUI
- Check **Enable IPsec Mobile Client Support**
- Set **User Authentication** to *Local Database*
- Check **Provide a virtual IP address to clients**
- Enter an **unused** private **Network** and appropriate subnet mask (such as */24*)
- Check **Provide a list of accessible networks to clients**
- Click **Save**

Phase 1

- Click the **Tunnels** Tab
- Click the **Create Phase1** button at the top if it appears, or edit the existing Mobile IPsec Phase 1
- Set **Key Exchange version** to *v2*
- Set **Authentication method** to *EAP-TLS*
- Set **My Identifier** to *Distinguished name* and enter in the hostname of the firewall

---

Note: This MUST match the Common Name of the server certificate!

- Set **Peer Identifier** to *User Distinguished name*, enter an e-mail address style identifier (e.g. *user@example.com*) – This isn't used, but is currently required by the GUI
- Select the server certificate created previously for **My Certificate**
- Select the appropriate CA for **My Certificate Authority**
- Set **Encryption algorithm** to *AES 256*
- Set **Hash algorithm** to *SHA256*
- Set **DH key group** to *2 (1024 bit)*
- Set **Lifetime** to *28800*
- Uncheck **Disable Rekey**
- Uncheck **Disable Reauth**
- Set **NAT Traversal** to *Auto*
- Check **Enable DPD**, set for *10* seconds and *5* retries
- Click **Save**

Phase 2

- Click  to show the Mobile IPsec Phase 2 list
- Click  to add a new Phase 2 entry if one does not exist, or click  to edit an existing entry
- Set **Mode** to **Tunnel IPv4**
- Set **Local Network** as desired
  - To pass all traffic, including Internet traffic, across the VPN, set the **Local Network** to *0.0.0.0/0*
- Enter an appropriate **Description**
- Set **Protocol** to *ESP*
- Set **Encryption algorithms** to ONLY *AES 256*
- Set **Hash algorithms** to ONLY *SHA1*
- Set **PFS Key Group** to *off*
- Set **Lifetime** to *3600*

- Click **Save**

34.30.3 Add Firewall Rules for IPsec

Firewall rules are necessary to pass traffic from IPsec clients.

- Navigate to **Firewall > Rules**, **IPsec** tab
- Review the current rules. If there is an "allow all" style rule, then there is no need to add another. Continue to the next task.
- Click  to add a new rule
- Set the **Protocol** to *any*, and set the **Source** and **Destination** to *any* as well
- Click **Save**
- Click **Apply Changes**

34.30.4 Import the CA to the Client PC

The server setup is complete, the following tasks will configure the client side.

- Export CA Cert from the pfSense router and download it to the client PC
  - Navigate to **System > Cert Manager**, **Certificate Authorities** tab in the pfSense webGUI
  - Click  by the CA to download *only* the certificate
- Locate the downloaded file on the client PC (e.g. *MyCA.crt*)
- Double click the CA file
- Click **Install Certificate...**
- Select *Local Machine*
- Click **Next**
- Click **Yes** at the UAC prompt if it appears
- Select *Place all Certificates in the following store*
- Click **Browse**
- Click *Trusted Root Certification Authorities*
- Click **Next**
- Click **Finish**
- Click **OK**
- Click **OK**

34.30.5 Import the Client Certificate to the Client PC

- Export client certificate from the pfSense router and download it to the client PC
  - Navigate to **System > Cert Manager**, **Certificates** tab in the pfSense webGUI
  - Click  by the certificate to download a `.p12` file containing the client certificate and key
- Locate the downloaded file on the client PC (e.g. *client1.p12*)
- Double click client certificate *.p12*
- Select *Current User*
- Click **Next**
- Click **Yes** at the UAC prompt if it appears
- Confirm the proper file is selected
- Click **Next**
- Click **Next**
- Click **Next**
- Click **Finish**
- Click **OK**

34.30.6 Add the Client VPN Connection

With the certificates properly imported, now it is time to create the client VPN connection. There are several ways to add such a connection, depending on the version of Windows being used. Adapt as needed.

- Open **Network and Sharing Center** on the client PC
- Click **Set up a new connection or network**
- Select *Connect to a workplace*
- Click **Next**
- Select *No, create a new connection*
- Click **Next**
- Click **Use my Internet Connection (VPN)**
- Enter the IP address or hostname of the server into the Internet address field

---

Note: This MUST match what is in the server certificate Common Name or a configured Subject Alternative Name!

- Enter a **Destination Name** to identify the connection
- Click **Create**

The connection has been added but with several undesirable defaults. For example the type defaults to automatic and it will latch onto a PPTP connection if one exists, which is very bad. So a few settings should be set by hand first:

- In Network Connection / Adapter Settings in Windows, find the connection created above

- Right click the connection
- Click **Properties**
- Click the **Security** tab
- Set **Type of VPN** to *IKEv2*
- Set **Data Encryption** to *Require Encryption (disconnect if server declines)*
- Set **Authentication / Use Extensible Authentication Protocol** to *Microsoft: Smart Card or other certificate (encryption enabled)*
- Click **Properties**
- Select *Use a certificate on this computer*
- Click *Advanced*
- Check **Certificate Issuer**
- Choose the imported CA Certificate (e.g. *myca*)
- Check **Extended Key Usage**
- Check **Client Authentication**
- Click **OK**
- Check **Verify the servers identity by validating the certificate**
- Check **Connect to these servers**
- Enter the pfSense hostname (same as in the CN of the server certificate!)
- Select the imported CA certificate (e.g. *myca*) in the **Trusted Root Certificate Authorities** box
- Uncheck *Use a different user name for the connection*
- Click **OK**

34.31 Connecting to Cisco PIX/ASA Devices with IPsec

Using IPsec to create a VPN tunnel between pfSense® router and a Cisco PIX should work OK. As always with IPsec, be sure that the Phase 1 and Phase 2 settings match up on both sides. If an acceptable transform set and policy are already in place, they may be used.

34.31.1 Example PIX IPsec Configuration

10.1.1.0	Network Behind the PIX
192.168.1.0	Network Behind the pfSense Router
1.2.3.4	IP of the pfSense router (WAN IP, CARP IP, etc)
dyn-map 10	Name of the existing crypto map (if any) and a unique ID for this tunnel. If there is an existing map, use its name, but a different number.
PFSVPN	Access list name for this connection.
(Pre-Shared Key)	PSK For this tunnel

```
access-list PFSVPN permit ip 192.168.1.0 255.255.255.0 10.1.1.0 255.255.255.0
access-list PFSVPN permit ip 10.1.1.0 255.255.255.0 192.168.1.0 255.255.255.0
access-list nonat permit ip 10.1.1.0 255.255.255.0 192.168.1.0 255.255.255.0
nat (inside) 0 access-list nonat
sysopt connection permit-ipsec
crypto ipsec transform-set 3dessha1 esp-3des esp-sha-hmac
crypto ipsec security-association lifetime seconds 86400 kilobytes 50000
crypto map dyn-map 10 ipsec-isakmp
crypto map dyn-map 10 match address PFSVPN
crypto map dyn-map 10 set pfs group2
crypto map dyn-map 10 set peer 1.2.3.4
crypto map dyn-map 10 set transform-set 3dessha1
crypto map dyn-map 10 set security-association lifetime seconds 86400 kilobytes␣
↪4608000
crypto map dyn-map interface outside
isakmp enable outside
isakmp key (Pre-Shared Key) address 1.2.3.4 netmask 255.255.255.255 no-xauth no-
↪config-mode
isakmp identity address
isakmp nat-traversal 20
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption 3des
isakmp policy 1 hash sha
isakmp policy 1 group 2
isakmp policy 1 lifetime 86400
```

34.31.2 Corresponding pfSense IPsec Config

In the above example, the pfSense IPsec tunnel should be set as follows:

Phase 1:

> **Remote Gateway**: (outside IP of the PIX) **Authentication Method**: *Pre-Shared Key* **Negotiation Mode**: *Main* **My Identifier**: *My IP Address* **Pre-Shared Key**: (The Pre-Shared Key) **Encryption Algorithm**: *3DES* **Hash Algorithm**: *SHA1* **DH Key Group**: *2* **Lifetime**: *86400* **NAT Traversal**: *Disable*

It may also be advisable to set **Proposal Checking** to *Obey* to avoid some issues with building a tunnel when the other side initiates.

Phase 2:

> **Mode**: *Tunnel IPv4* **Local Network**: *LAN Subnet* **Remote Network**: *10.1.1.0/24* **Protocol**: *ESP* **Encryption Algorithm**: *3DES* (others may also be checked, but be sure to leave 3DES checked) **Hash Algorithm**: *SHA1* **PFS Key Group**: *2* **Lifetime**: *86400*

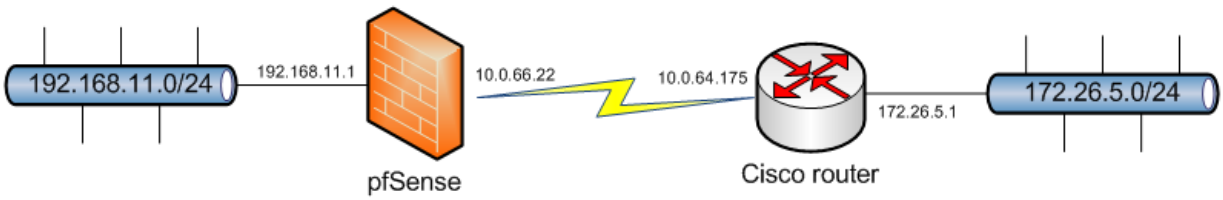
Rules must also be added to **Firewall > Rules** on the **IPsec** Tab to permit traffic from *10.1.1.0/24* (or specific IPs/Alias) to the LAN Net (or specific IPs/Alias).

34.32 Connecting to Cisco IOS Devices with IPsec

This page describes how to configure IPsec to connect pfSense® router and a Cisco IOS router with IPsec capabilities.

34.32.1 Example Network

This diagram shows the specifics of the network where this VPN is being configured. For the sake of this documentation, both hosts were on private subnets, but functionally equivalent to two hosts across the Internet.



34.32.2 Configuring the router

First, configure the phase 1 settings with a crypto isakmp policy. The following sets it for 3DES, SHA and group 2 to match the pfSense configuration shown later.

```
crypto isakmp policy 10
 encr 3des
 authentication pre-share
 group 2
```

Next, configure the pre-shared key. The key in this example is ABCDEFG, but be sure to use something random and secure for any production deployments. *10.0.66.22* is the WAN IP of the pfSense system being used.

```
crypto isakmp key ABCDEFG address 10.0.66.22 no-xauth
```

Next configure the transform set for phase 2. This uses *ESP*, *3DES* and *SHA*. The transform set is named *3DES-SHA*, which is how it will be referred to later.

```
crypto ipsec transform-set 3DES-SHA esp-3des esp-sha-hmac
```

Now configure an access list that will match the local and remote subnets on the pfSense router. This is configured as *access-list 100*, which will be used in the next step. Remember this uses wildcard masks, so a /24 network (255.255.255.0 mask) is represented as *0.0.0.255*.

```
access-list 100 permit ip 192.168.11.0 0.0.0.255 172.26.5.0 0.0.0.255
access-list 100 permit ip 172.26.5.0 0.0.0.255 192.168.11.0 0.0.0.255
```

Now configure the crypto map for this VPN:

```
crypto map PFSVPN 15 ipsec-isakmp
 set peer 10.0.66.22
 set transform-set 3DES-SHA
 set pfs group2
 match address 100
```

Lastly, under the interface configuration for the interface where the VPN will terminate (the one with the public IP), assign the crypto map:

```
interface FastEthernet0/0
crypto map PFSVPN
```

The configuration is then finished on the Cisco side.

34.32.3 Configuring pfSense Software

This screenshot shows the pfSense configuration matching the above Cisco configuration.

In the above example, the pfSense IPsec tunnel should be set as follows:

Phase 1:

> **Remote Gateway**: *10.0.64.175* **Authentication Method**: *Pre-Shared Key* **Negotiation Mode**: *Main* **My Identifier**: *My IP Address* **Pre-Shared Key**: *ABCDEFG* **Encryption Algorithm**: *3DES* **Hash Algorithm**: *SHA1* **DH Key Group**: *2* **Lifetime**: *28800* **NAT Traversal**: *Disable*

It may also be advisable to set **Proposal Checking** to *Obey* to avoid some issues with building a tunnel when the other side initiates.

Phase 2:

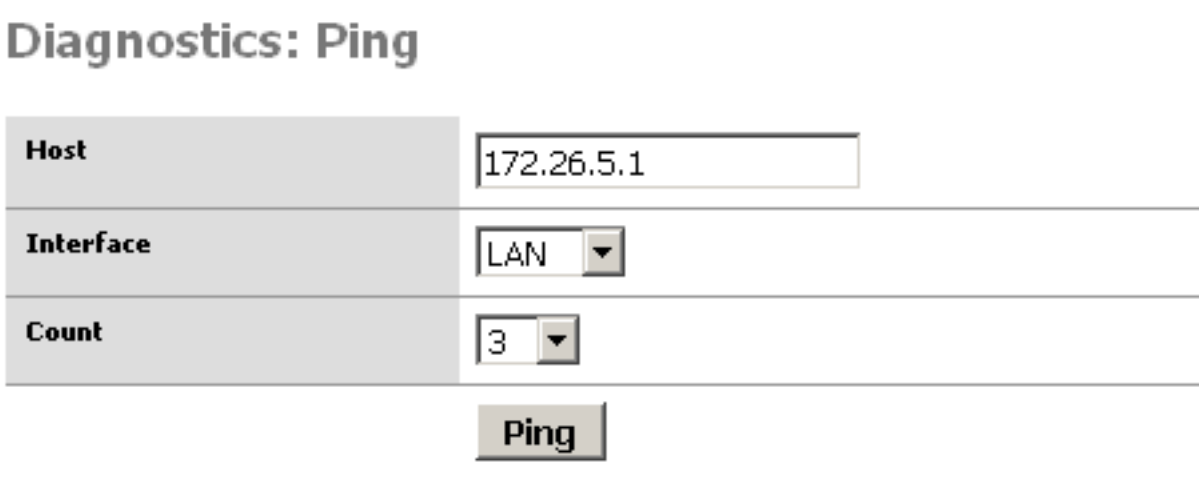
> **Mode**: *Tunnel IPv4* **Local Network**: *LAN Subnet* **Remote Network**: *172.26.5.0/24* **Protocol**: *ESP* **Encryption Algorithm**: *3DES* (others may also be checked, but be sure to leave 3DES checked) **Hash Algorithm**: *SHA1* **PFS Key Group**: *2* **Lifetime**: *3600*

34.32.4 Testing the connection

To test the connection, from the pfSense router, do the following:

- Navigate to **Diagnostics > Ping**
- Enter an IP address on the remote network
- Choose the LAN interface
- Click Ping.

The initial negotiation may make all three of the first pings timeout, so try it a second time as well. If configured as depicted above, once the tunnel connects, the following will be seen:



34.32.5 Troubleshooting

If the connection doesn't come up, there is a mismatch somewhere in the configuration. Depending on specifics, more useful information may be obtained from pfSense router or the Cisco router. Checking logs on both ends is recommended. For pfSense software, browse to **Status > System Logs** on the **IPsec** tab. For Cisco, run **debug crypto isakmp** and **term mon** (if not connected via serial console) to make the debug messages appear in a session. The output can be verbose, but will usually tell specifically what was mismatched.

"No NAT" List on Cisco IOS

It may also be necessary to tell Cisco IOS *not* to NAT the traffic that is destined for the IPsec tunnel. There are several ways to accomplish this, depending on how the router has NAT configured. If the following example does not help, there are several examples that turn up in a Google search for "cisco ios nonat ipsec":

```
ip nat inside source route-map NONAT interface FastEthernet0/0 overload
access-list 110 deny   ip 172.26.5.0 0.0.0.255 192.168.11.0 0.0.0.255
access-list 110 permit ip 172.26.5.0 0.0.0.255 any
route map NONAT permit 10
 match ip address 110
```

This will direct the router to prevent NAT if the traffic is going from the subnet behind the Cisco router to the subnet behind the pfSense router, but allow it in all other cases.

34.33 IPsec Site-to-Site VPN Example with Pre-Shared Keys

A site-to-site IPsec tunnel interconnects two networks as if they were directly connected by a router. Systems at Site A can reach servers or other systems at Site B, and vice versa. This traffic may also be regulated via firewall rules, as with any other network interface. If more than one client will be connecting to another site from the same controlled location, a site-to-site tunnel will likely be more efficient, not to mention more convenient and easier to support.

With a site-to-site tunnel, the systems on either network need not have any knowledge that a VPN exists. No client software is needed, and all of the tunnel work is handled by the tunnel endpoints. This is also a good solution for devices that have network support but do not handle VPN connections such as printers, cameras, HVAC systems, and other embedded hardware.

See also:

- *IPsec*
- *IPsec Configuration*
- *Troubleshooting Duplicate IPsec SA Entries*
- *Client Routing and Gateway Considerations*
- *Accessing Firewall Services over IPsec VPNs*

34.33.1 Site-to-site example configuration

The key to making a working IPsec tunnel is to ensure that both sides have matching settings for authentication, encryption, and so on. Before starting, make a note of the local and remote WAN IP addresses, as well as the local and remote internal subnets that will be carried across the tunnel. Aside from the cosmetic tunnel **Description** and these pieces of information, the other connection settings will be identical.

The following settings are assumed by this example and some of the subsequent examples in the IPsec recipes:

Table 9: IPsec Endpoint Settings

Site A		Site B	
Name	Austin Office	Name	London Office
WAN IP	198.51.100.3	WAN IP	203.0.113.5
LAN Subnet	10.3.0.0/24	LAN Subnet	10.5.0.0/24
LAN IP	10.3.0.1	LAN IP	10.5.0.1

Note: To avoid issues with security association duplication, this example uses settings described in *Troubleshooting Duplicate IPsec SA Entries*.

Figure *Site-to-Site IPsec* shows the general layout of this VPN.

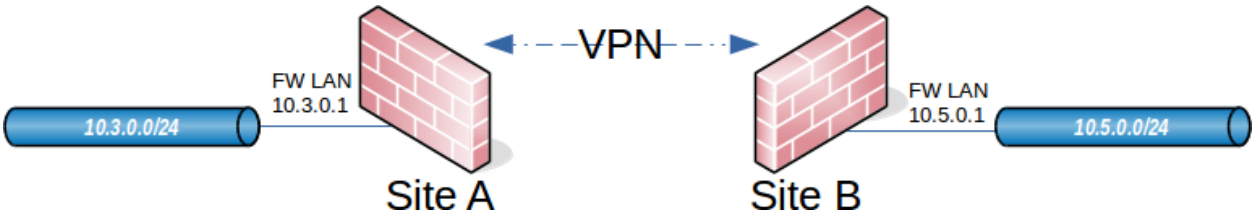


Fig. 43: Site-to-Site IPsec

34.33.2 Site A

Start with configuring the tunnel and related settings on the firewall at Site A.

Phase 1

To add a new IPsec phase 1:

- Navigate to **VPN > IPsec**
- Click  **Add P1**
- Fill in the settings as described below
- Click **Save** when complete

Use the following settings for the phase 1 configuration. Many of these settings may be left at their default values unless otherwise noted.

See also:

For comprehensive coverage of all IPsec phase 1 settings, see *Phase 1 Settings*.

First, fill in the top section that holds the general phase 1 information, shown in Figure *figure-vpn-tunnel-settings*. Items in bold are required. Fill in the settings as described:

Disabled Uncheck this box so that the tunnel will be operational.

Key Exchange version Specifies whether to use IKEv2 or IKEv1. IKEv2 is the best practice when supported by both endpoints. If one side does not support IKEv2, use IKEv1 instead.

Internet Protocol *IPv4* in most cases unless both WANs have IPv6, in which case either type may be used.

Interface Most likely set to *WAN* , but see the note at *Interface Selection* on selecting the proper interface when unsure.

Remote Gateway The WAN address at Site B, `203.0.113.5` in this example.

Description Text describing the purpose or identity of the tunnel. The best practice is to put the name of Site B in this box, and brief detail about the purpose of the tunnel to help with future administration.

For this example `ExampleCo London Office` is used for the **Description** to identify where this example tunnel terminates.

The next section controls IPsec phase 1 proposals, also referred to as Authentication. The defaults are desirable for most of these settings, and simplifies the process.

Authentication Method The default, *Mutual PSK*, is used for this example.

My Identifier The default, *My IP Address*, is kept for this example.

Peer Identifier The default, *Peer IP Address*, is kept for this example.

Pre-Shared Key Use a strong key, at least 10 characters in length containing a mix of upper and lower-case letters, numbers and symbols. Enter a custom key or click  **Generate new Pre-Shared Key** to automatically populate the field with a random long string suitable for use as a Pre-Shared Key.

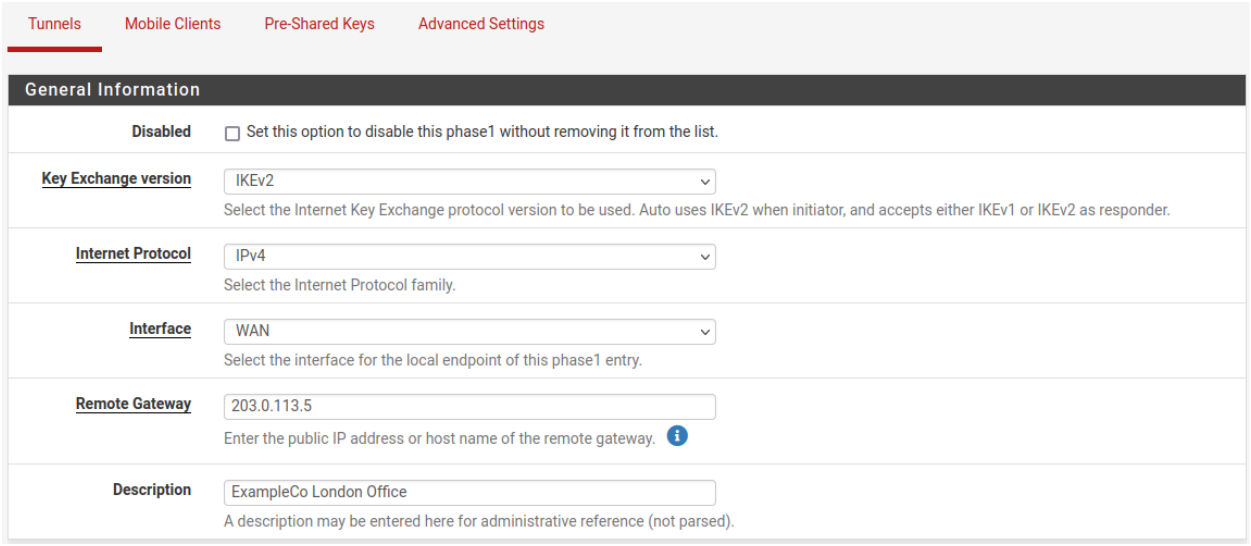


Fig. 44: Site A IPsec Phase 1 General Settings

> **Warning:** This is the most important setting to get correct. As mentioned in the VPN overview, IPsec using pre-shared keys can be broken if the tunnel uses a weak key.

The **exact** same key must be entered into the tunnel configuration for Site B later, so note it down or copy and paste it elsewhere. Copy and paste may come in handy, especially with a complex key.

Encryption Algorithm Use *AES* with a key length of *256 bits*.

Hash Algorithm Use *SHA256* if both sides support it, otherwise use the strongest hash supported by both endpoints.

DH Group The default of *14 (2048 bit)* is OK, higher values are more secure but use more CPU.

Lifetime The default `28800` is OK for this endpoint.

See also:

See *Troubleshooting Duplicate IPsec SA Entries* for best practices in choosing lifetime values.

The other lifetime-related values should be left at their defaults on this endpoint as they are automatically calculated as the correct values.

Child SA Close Action Set this endpoint to **Restart/Reconnect** so that the phase 2 entries will be reconnected if they get disconnected.

Dead Peer Detection Leave checked, the default **Delay** of `10` seconds and **Max Failures** of `5` is adequate. Depending on the needs at a site a higher value may be better, such as `30` seconds and `6` retries, but a problematic WAN connection on either side may make that too low.

Click **Save** as mentioned above to complete the phase 1 setup.

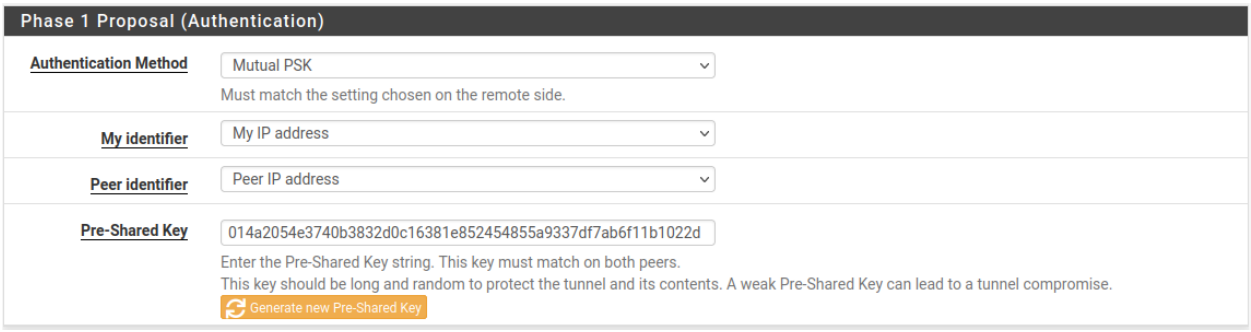


Fig. 45: Site A Phase 1 Authentication Settings

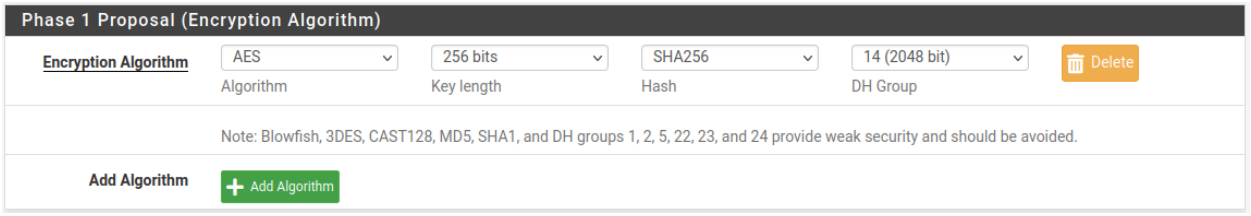


Fig. 46: Site A Phase 1 Encryption Settings

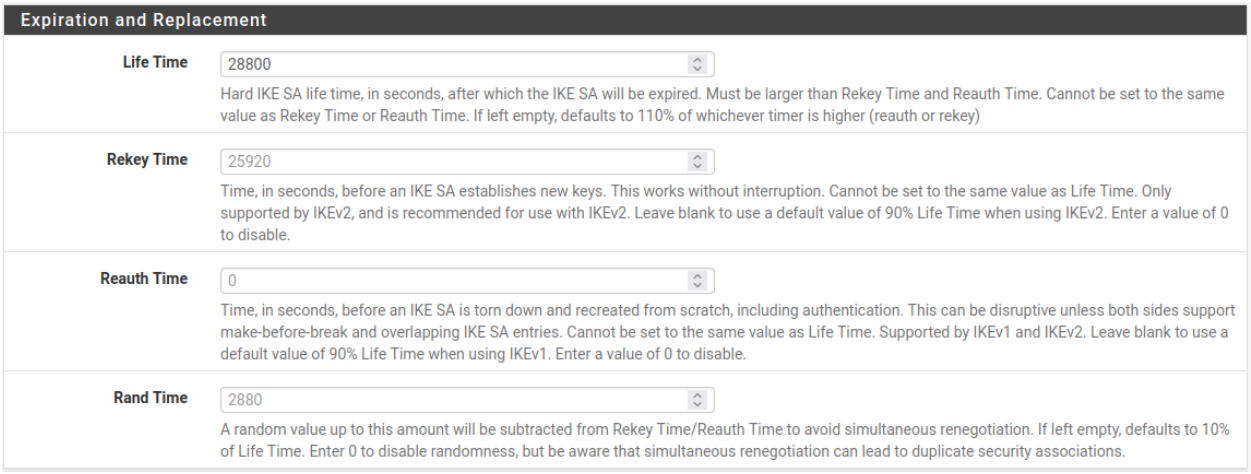


Fig. 47: Site A Phase 1 Lifetime Settings

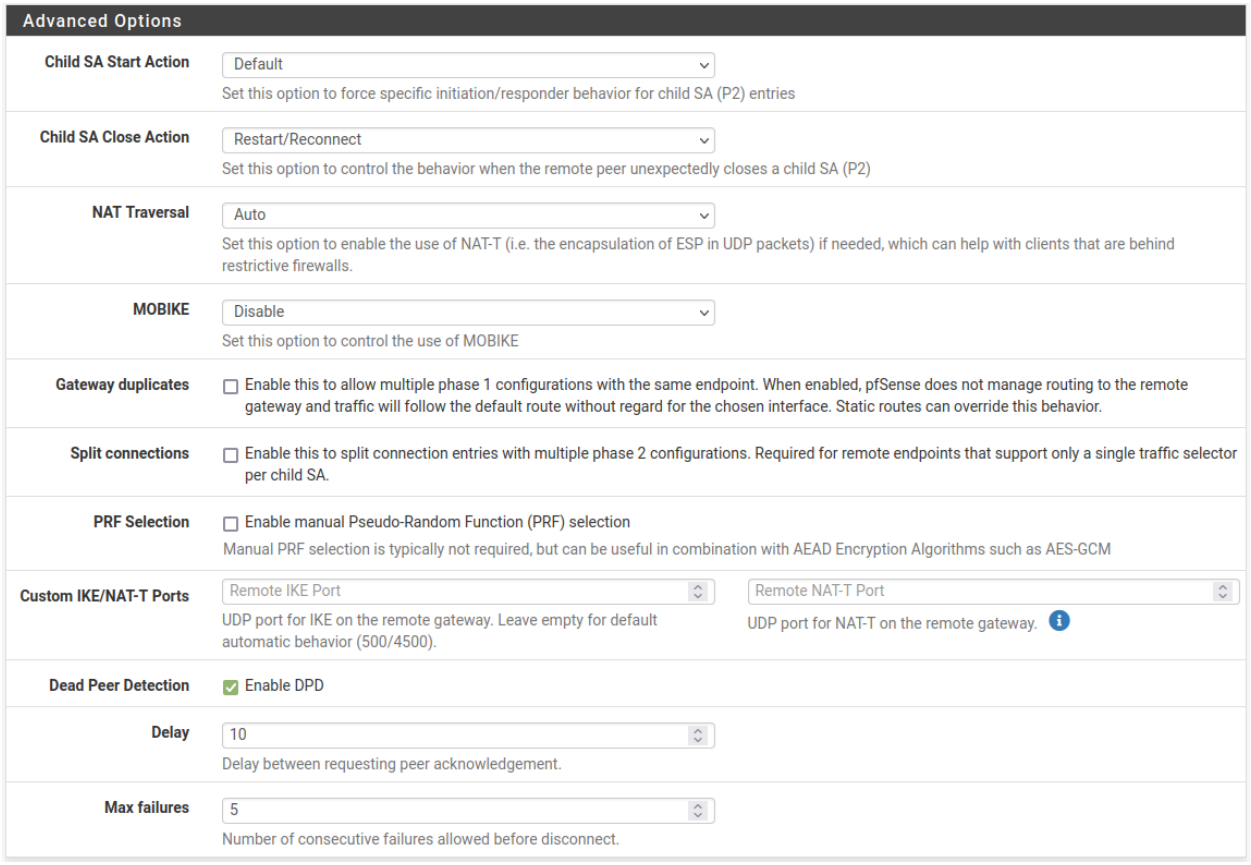


Fig. 48: Site A Phase 1 Advanced Settings

Phase 2

With the phase 1 entry complete, now a new phase 2 definition to the VPN:

- Click  **Show Phase 2 Entries** as seen in Figure *Site A Phase 2 List (Empty)* to expand the phase 2 list for this VPN.
- Click  **Add P2** to add a new phase 2 entry, as seen in Figure *Adding a Phase 2 entry to Site A*.

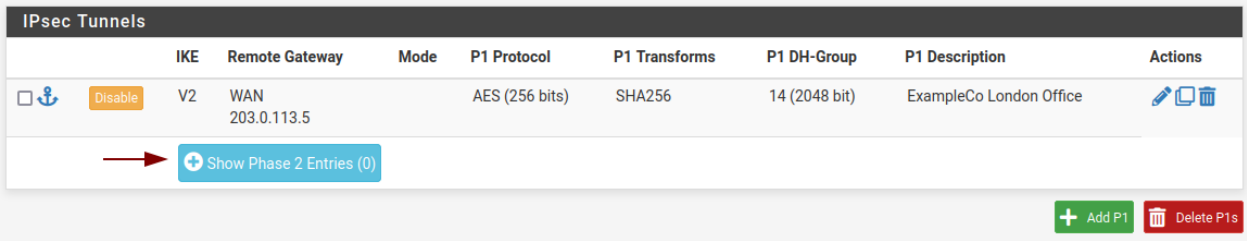


Fig. 49: Site A Phase 2 List (Empty)

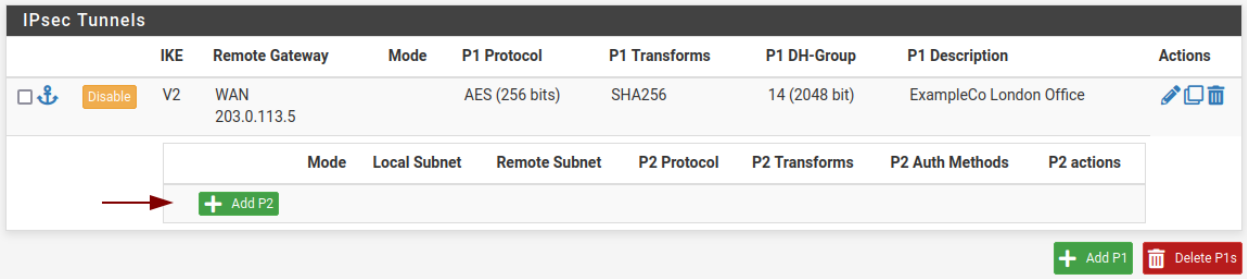


Fig. 50: Adding a Phase 2 entry to Site A

Now add settings for phase 2 on this VPN. The settings for phase 2 (Figure *Site A Phase 2 General Settings*) can vary more than phase 1.

See also:

For comprehensive coverage of all IPsec phase 2 settings, see *Phase 2 Settings*.

Mode Since this example is for a policy-based tunnel, select *Tunnel IPv4*

Local Network In most cases the best practice is to leave this as *LAN Subnet*, but it can be changed to *Network* with the proper subnet value filled in. In this case that would be `10.3.0.0/24`. Leaving it as *LAN Subnet* will ensure that if the network is renumbered in the future, this end of the tunnel will follow. If that does happen, the other end must be changed manually.

NAT/BINAT Set to *None*.

Remote Network Set to the network at Site B, in this case `10.5.0.0/24`.

Description A brief description of the network(s) involved in this phase 2 entry.

The remainder of the phase 2 settings cover the encryption of the traffic. Encryption algorithms and Hash algorithms can both be set to allow multiple options in phase 2, and both sides will negotiate and agree upon the settings so long as each side has at least one of each in common. In some cases that may be a good thing, but it is usually better to restrict this to the single specific options desired on both sides.

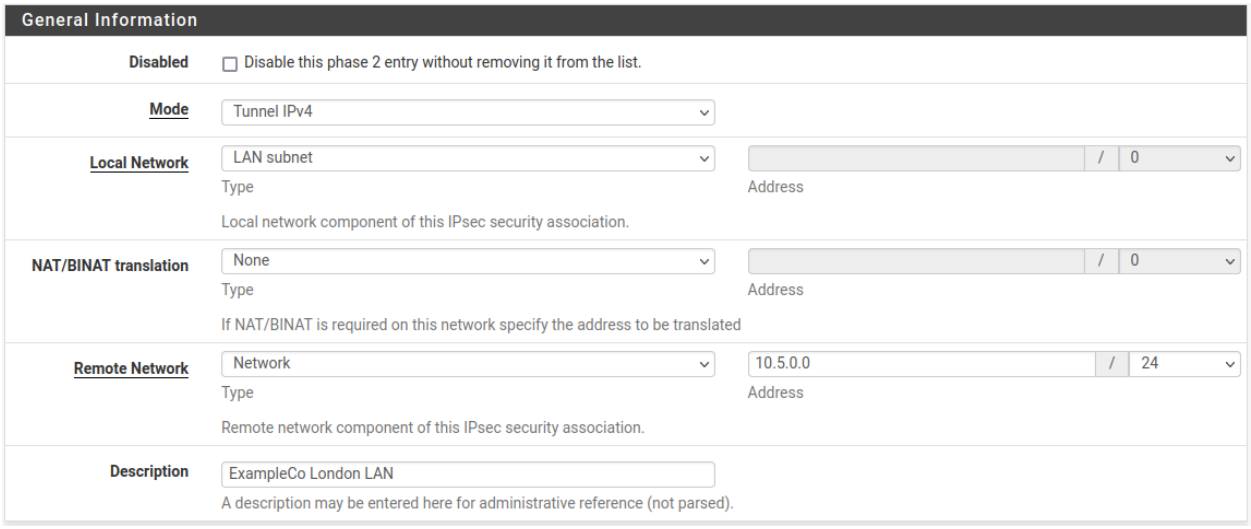


Fig. 51: Site A Phase 2 General Settings

Protocol Set to *ESP* for encryption.

Encryption algorithm The best practice is to use an AEAD cipher such as AES-GCM if it is supported by both endpoints.

Select *AES256-GCM* with a *128 bit* key length. Otherwise, use *AES 256*, or the highest strength cipher supported by both endpoints.

Hash algorithm If AES-GCM is selected for **Encryption Algorithm**, do not select any hashes.

Otherwise, use *SHA256* or whichever hash supported by both sides is strongest.

PFS Perfect Forward Secrecy can help protect against certain key attacks, but is optional. This example uses *14 (2048 bit)*.

Lifetime Use `3600` for this example, and leave **Rekey Time** and **Rand Time** at their default calculated placeholder values.

To finalize the settings and put them into action: * Click *Save* * Click *Apply Changes* on the IPsec Tunnels screen, as seen in Figure *Apply IPsec Settings*.

Firewall Rules

The tunnel for Site A is finished, but now firewall rules are needed to allow traffic from the network at Site B to enter through the IPsec tunnel.

Navigate to **Firewall > Rules** on the **IPsec** tab and add rules there to pass traffic from the remote side of the VPN.

See also:

See *Firewall* for specifics on adding rules.

Rules may be as permissive as desired, (allow any protocol from anywhere to anywhere), or restrictive (allow TCP from a certain host on Site B to a certain host at Site A on a certain port).

As with other firewall rules, the connections are checked on the way *into* the firewall, so the source of all traffic on the IPsec tab rules will be remote VPN networks, such as those at Site B.

Make sure the Source address(es) on the firewall rules match Site B addresses, such as `10.5.0.0/24` . The destination addresses will be on Site A, such as `10.3.0.0/24`.

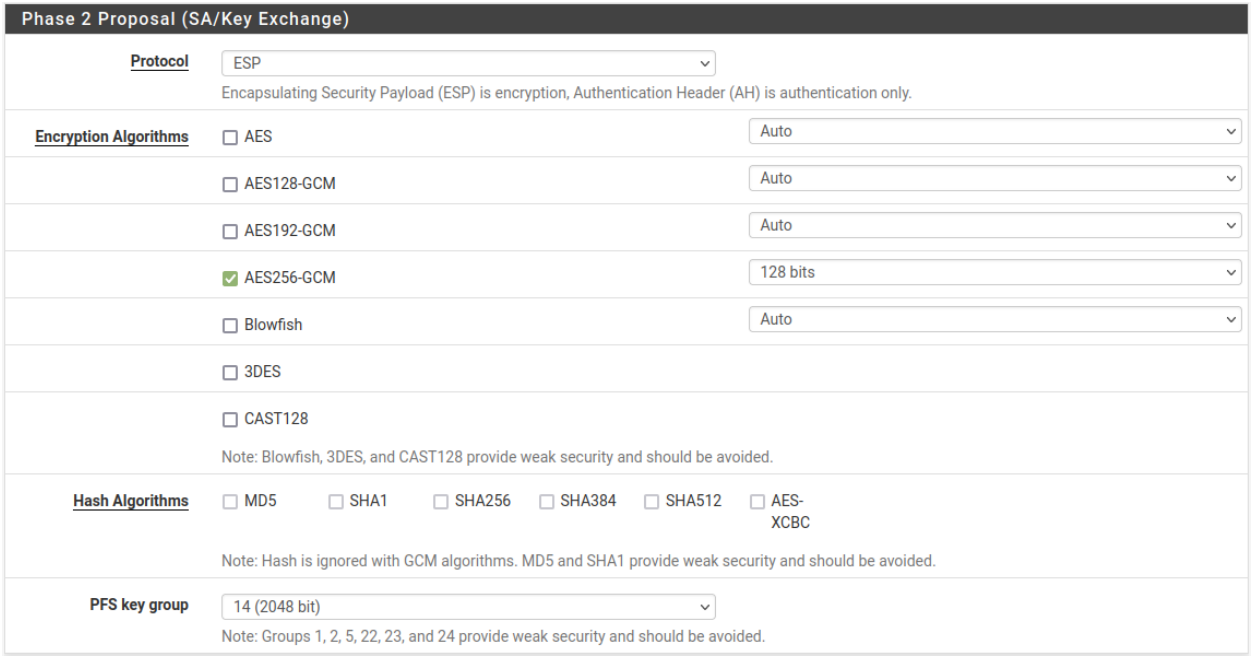


Fig. 52: Site A Phase 2 Proposal Settings

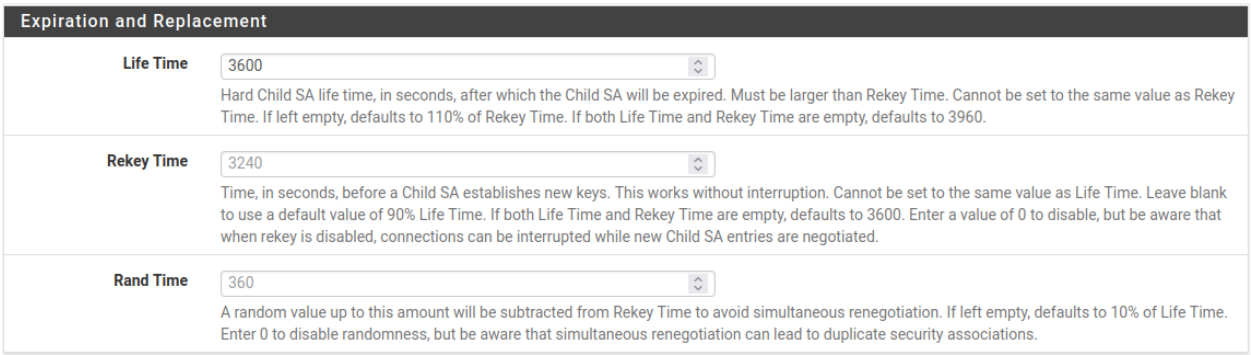


Fig. 53: Site A Phase 2 Lifetime Settings

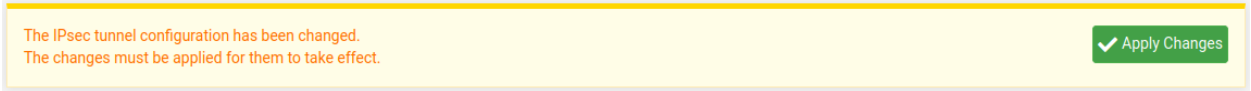


Fig. 54: Apply IPsec Settings

34.33.3 Site B

Now that Site A is configured, it is time to tackle Site B. Repeat the process on the Site B endpoint to add a tunnel.

Only a few parts of this setup will differ from Site A as shown in Figure *Site B Phase 1 General Settings* and Figure *Site B Phase 2 General Settings*:

- The phase 1 settings for WAN address, Description, Life Time, Child SA Start Action, and Child SA Close Action
- The phase 2 tunnel networks and Life Time

Add a phase 1 entry to the Site B firewall using identical settings used on Site A but with the following differences:

Remote Gateway The WAN address at Site A, `198.51.100.3`.

Description `ExampleCo Austin Office`.

Life Time At least 10% higher than Site A, `31680`

Child SA Start Action Set to **None (Responder Only)** so that this endpoint will not initiate on its own, but will wait for Site A to initiate.

Child SA Close Action Set this endpoint to **Close Connection and clear SA** so that the phase 2 will not automatically reconnect, since Site A will be managing that.

- Click **Save**

Add a phase 2 entry to the Site B firewall using identical settings used on Site A but with the following differences.

Remote Subnet The network at Site A, in this case `10.3.0.0/24`.

Description `ExampleCo Austin LAN`.

Life Time At least 10% higher than Site A, `5400`

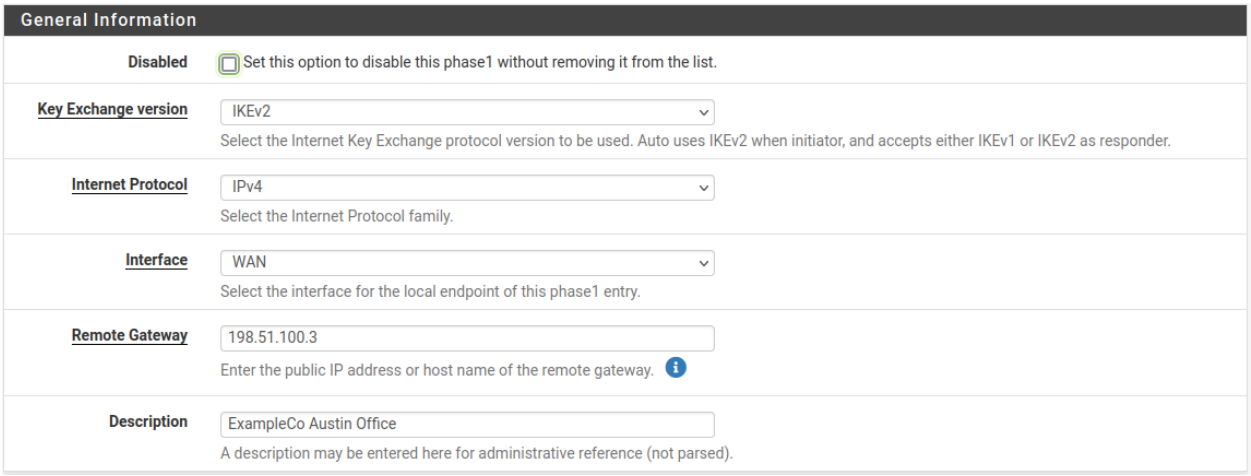


Fig. 55: Site B Phase 1 General Settings

- Click **Save**
- Click **Apply changes** on the IPsec Tunnels screen.

As with Site A, firewall rules must also be added to allow traffic on the tunnel to cross from Site A to Site B. Add these rules to the **IPsec** tab under **Firewall > Rules**. For more details, see *IPsec and firewall rules*. This time, the source of the traffic would be Site A, destination Site B.

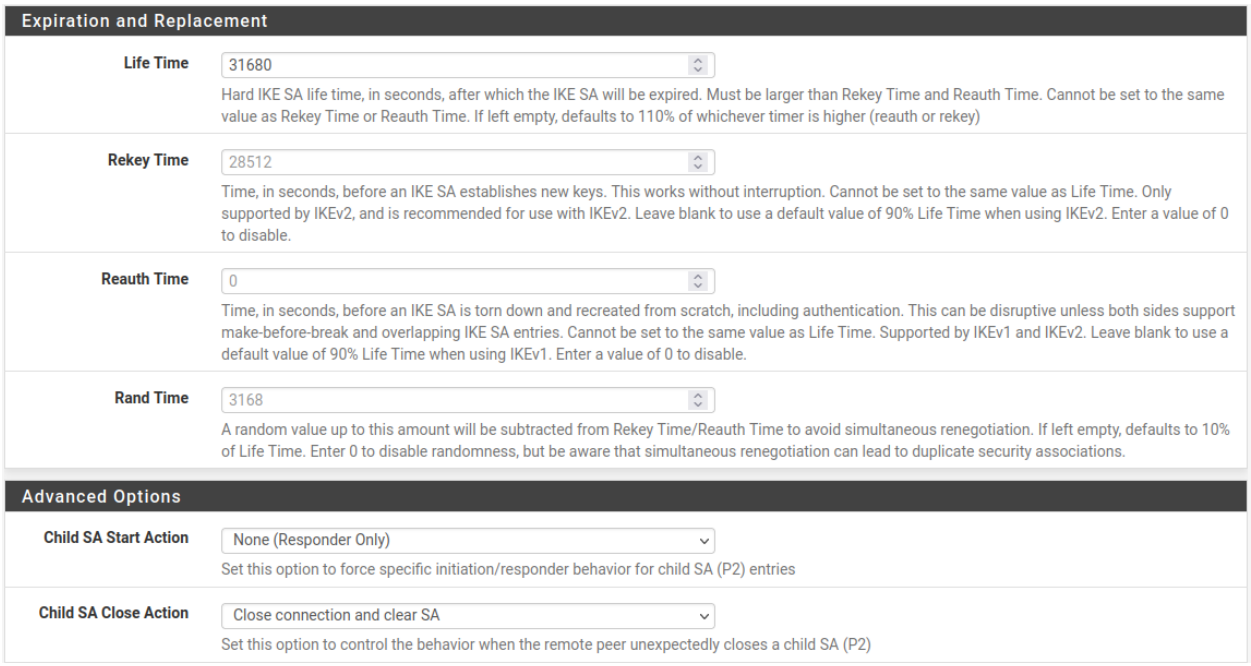


Fig. 56: Site B Phase 1 Other Settings

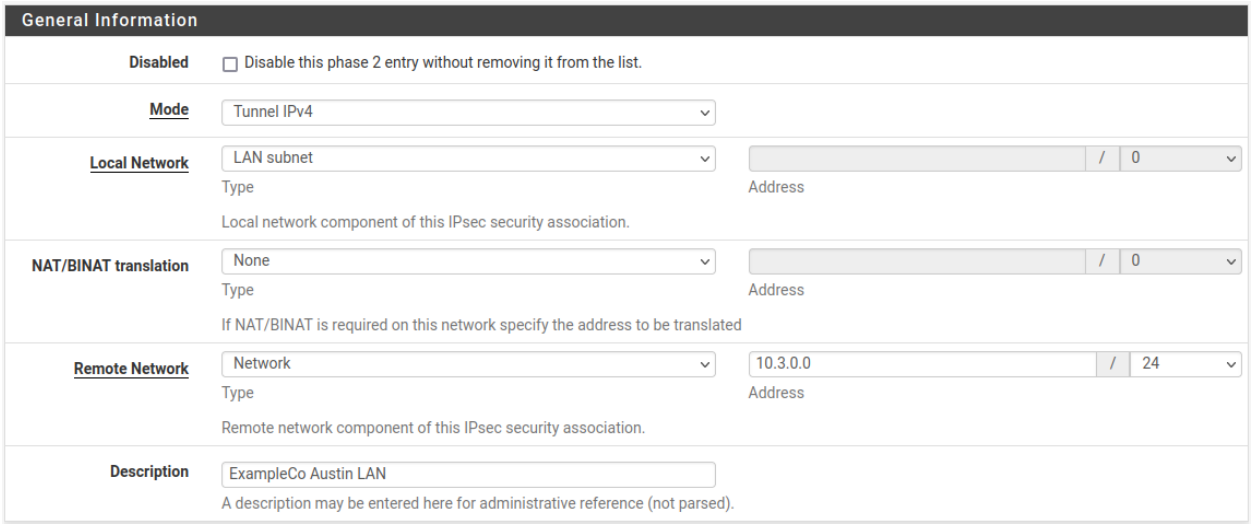


Fig. 57: Site B Phase 2 General Settings

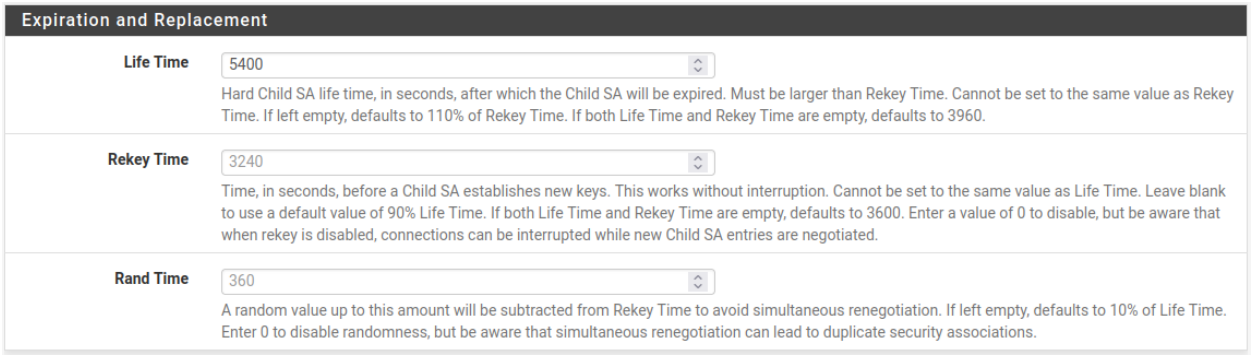


Fig. 58: Site B Phase 2 Lifetime Settings

34.33.4 Check Status

Both tunnels are now configured and active. Check the IPsec status by visiting **Status > IPsec**. A description of the tunnel is shown along with its status.

If the tunnel is not listed as **Established**, there may be a problem establishing the tunnel. This soon, the most likely reason is that no traffic has attempted to cross the tunnel.

A connect button is offered on this screen that will attempt to initiate the tunnel. Click the  **Connect VPN** button to attempt to bring up the tunnel, as seen in Figure *Site A IPsec Status*.

If the connect button does not appear, try to ping a system in the remote subnet at Site B from a device inside of the phase 2 local network at Site A (or vice versa) and see if the tunnel establishes. Look at *Testing IPsec Connectivity* for other means of testing a tunnel.

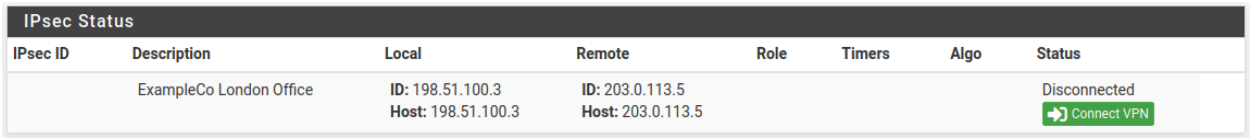
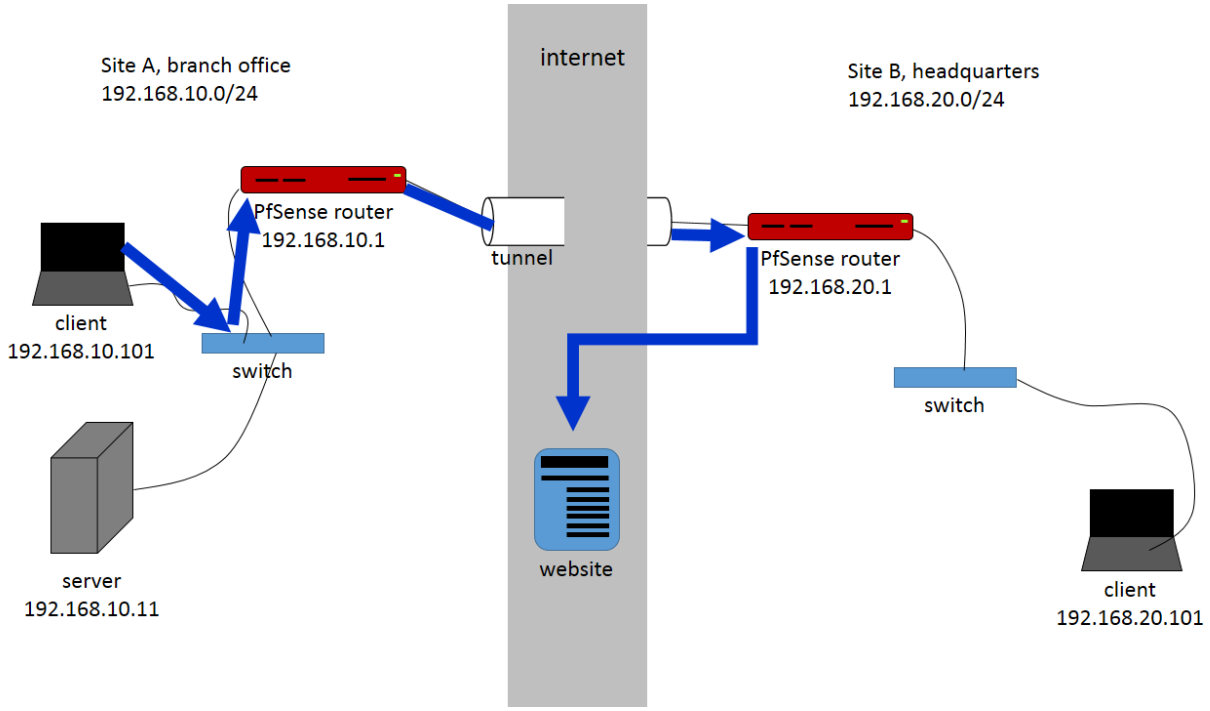


Fig. 59: Site A IPsec Status

Failing that, the IPsec logs will typically offer an explanation. They are located under **Status > System Logs** on the **IPsec** tab. Be sure to check the status and logs at both sites. For more troubleshooting information, check the *Troubleshooting IPsec VPNs* section later in this chapter.

34.34 Routing Internet Traffic Through a Site-to-Site IPsec Tunnel

It is possible to use IPsec on a pfSense® router to send Internet traffic from Site A such that it would appear to be coming from Site B. This may be needed if a vendor requires that connections originate from a specific address at Site B.



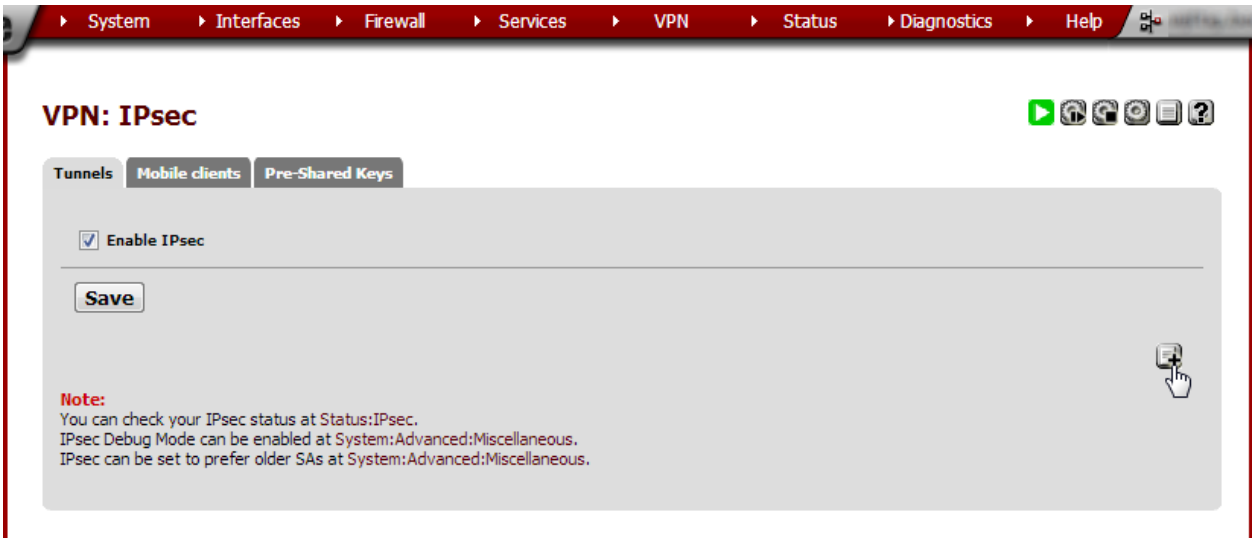
In this article we have two sites:

1. **Site A is a branch office**, LAN subnet 192.168.10.0/24
2. **Site B is the main office** through which all Internet traffic is routed, 192.168.20.0/24

34.34.1 Set up the IPsec tunnel Phase 1

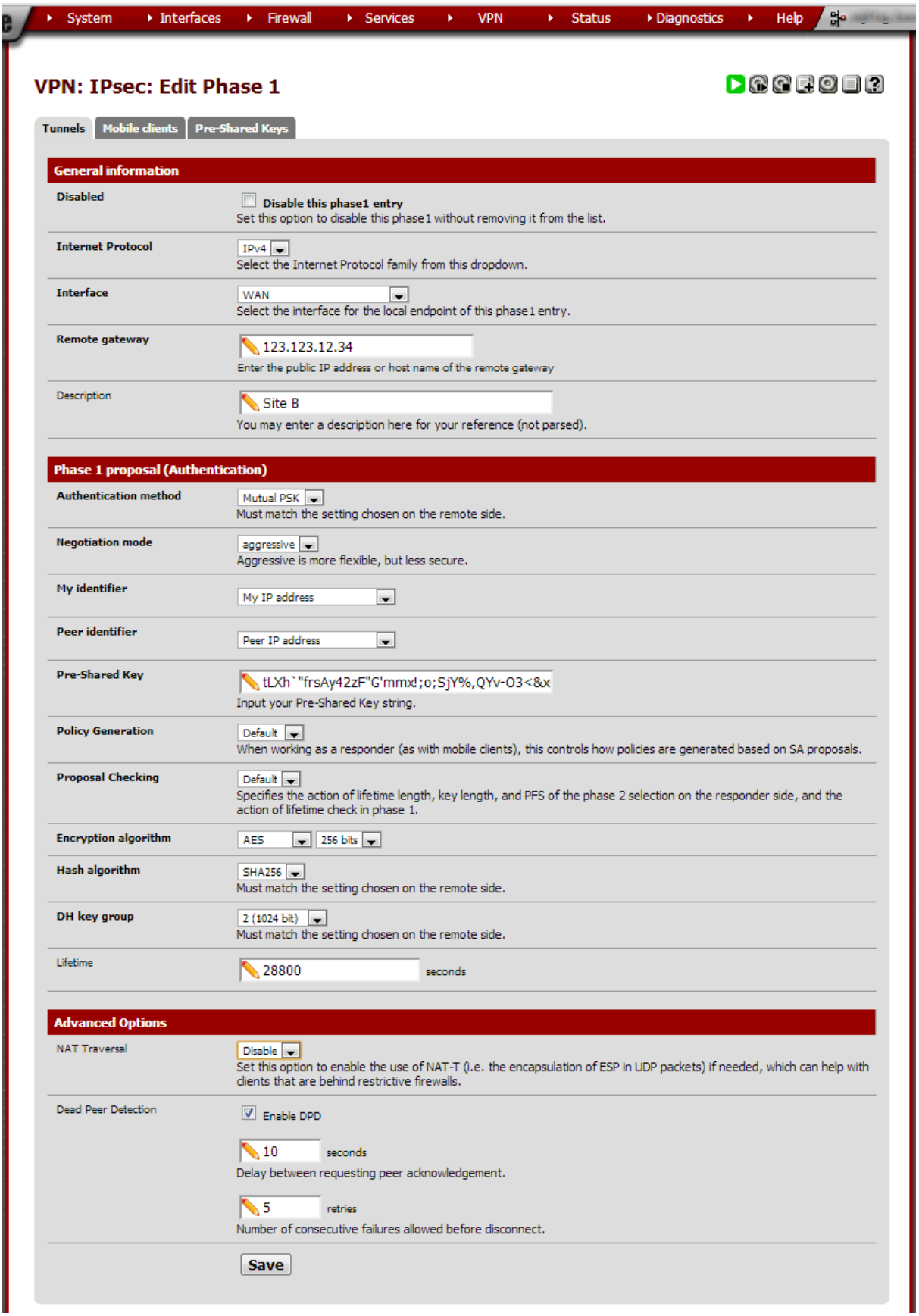
Site A Configuration

In the **VPN** menu select **IPsec**. It opens on the **Tunnels tab**. Click the **+** button to create a new Phase 1 setup. (Make sure **Enable IPsec** is checked and saved.)

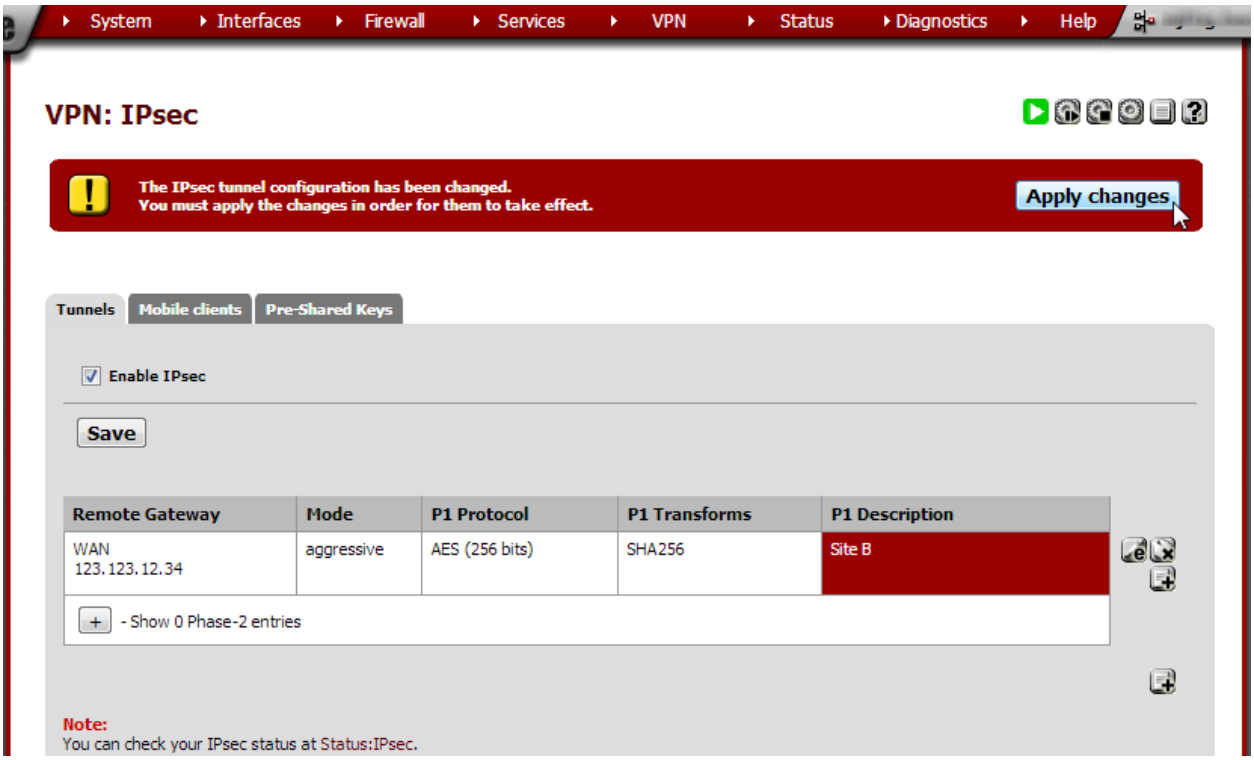


Enter these values:

Field	Value	Notes
Internet Protocol	IPv4	
Interface	WAN	Unless using a separate OPT interface
Description	Site B	The site's locality or another suitable description
Authentication method	Mutual PSK	
Negotiation mode	aggressive	
My identifier	My IP address	
Peer identifier	Peer IP address	
Pre-Shared Key	A long key.	This can be generated using external utilities but be careful to copy it without extra spaces.
Policy Generation	Default	
Proposal Checking	Default	
Encryption algorithm	AES 256bits	Read this comparison of encryption algorithms.
Hash algorithm	SHA256	Read this comparison of hash algorithms.
DH key group	2 (1024 bit)	Read this explanation of Perfect forward secrecy.
Lifetime	28800	
NAT Traversal	Disable	Turn this off unless it is definitely needed.
Dead Peer Detection	Enable: 10 seconds, 5 retries	Leave this on unless the other side does not properly support DPD.


System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
VPN: IPsec: Edit Phase 1
Tunnels
Mobile clients
Pre-Shared Keys
General information
Disabled
Disable this phase1 entry
Set this option to disable this phase1 without removing it from the list.
Internet Protocol
IPv4
Select the Internet Protocol family from this dropdown.
Interface
WAN
Select the interface for the local endpoint of this phase1 entry.
Remote gateway
123.123.12.34
Enter the public IP address or host name of the remote gateway
Description
Site B
You may enter a description here for your reference (not parsed).
Phase 1 proposal (Authentication)
Authentication method
Mutual PSK
Must match the setting chosen on the remote side.
Negotiation mode
aggressive
Aggressive is more flexible, but less secure.
My identifier
My IP address
Peer identifier
Peer IP address
Pre-Shared Key
tLXh`"frsAy42zF"G'mmx!;o;SjY%,QYv-O3<&x
Input your Pre-Shared Key string.
Policy Generation
Default
When working as a responder (as with mobile clients), this controls how policies are generated based on SA proposals.
Proposal Checking
Default
Specifies the action of lifetime length, key length, and PFS of the phase 2 selection on the responder side, and the action of lifetime check in phase 1.
Encryption algorithm
AES
256 bits
Hash algorithm
SHA256
Must match the setting chosen on the remote side.
DH key group
2 (1024 bit)
Must match the setting chosen on the remote side.
Lifetime
28800
seconds
Advanced Options
NAT Traversal
Disable
Set this option to enable the use of NAT-T (i.e. the encapsulation of ESP in UDP packets) if needed, which can help with clients that are behind restrictive firewalls.
Dead Peer Detection
Enable DPD
10
seconds
Delay between requesting peer acknowledgement.
5
retries
Number of consecutive failures allowed before disconnect.
Save

Note that the Phase 1 entry is now shown on the IPsec page. Click **Save** and in the next screen click **Apply Changes**.



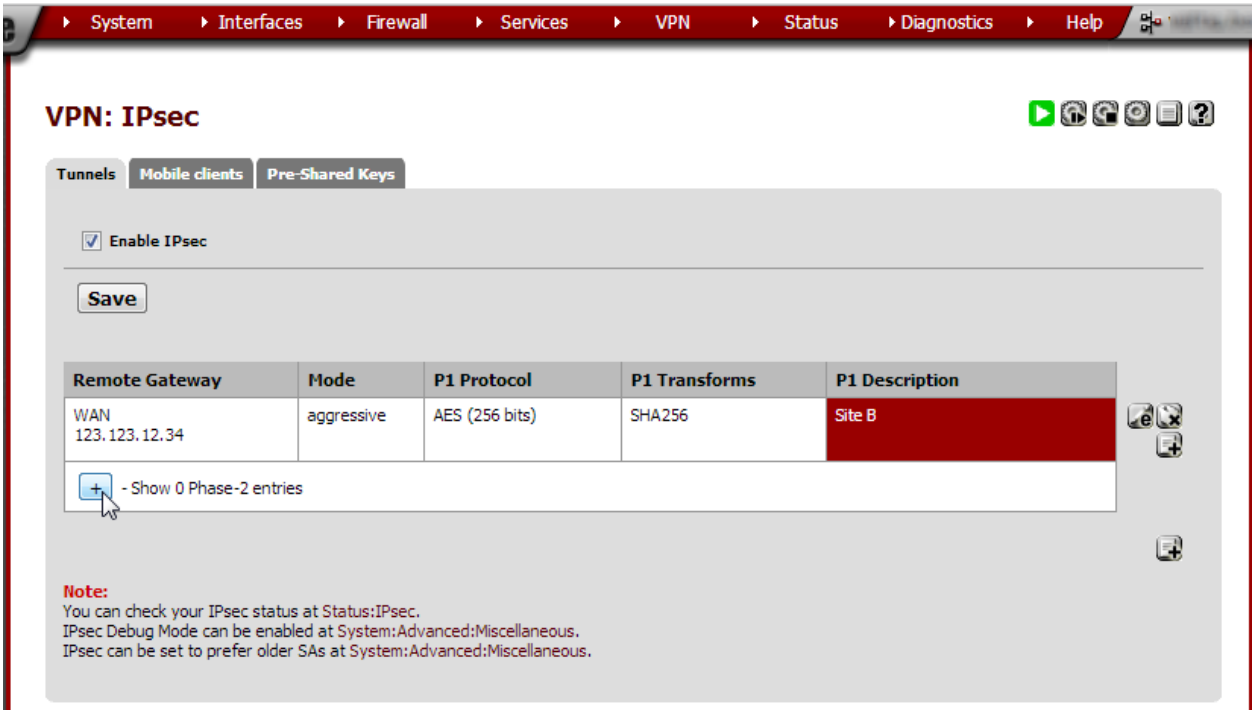
Site B Configuration

Do the same as in Site A but in the **Remote Gateway** field enter Site A's public IP address or FQDN and in the **Description** field enter 'Site A'.

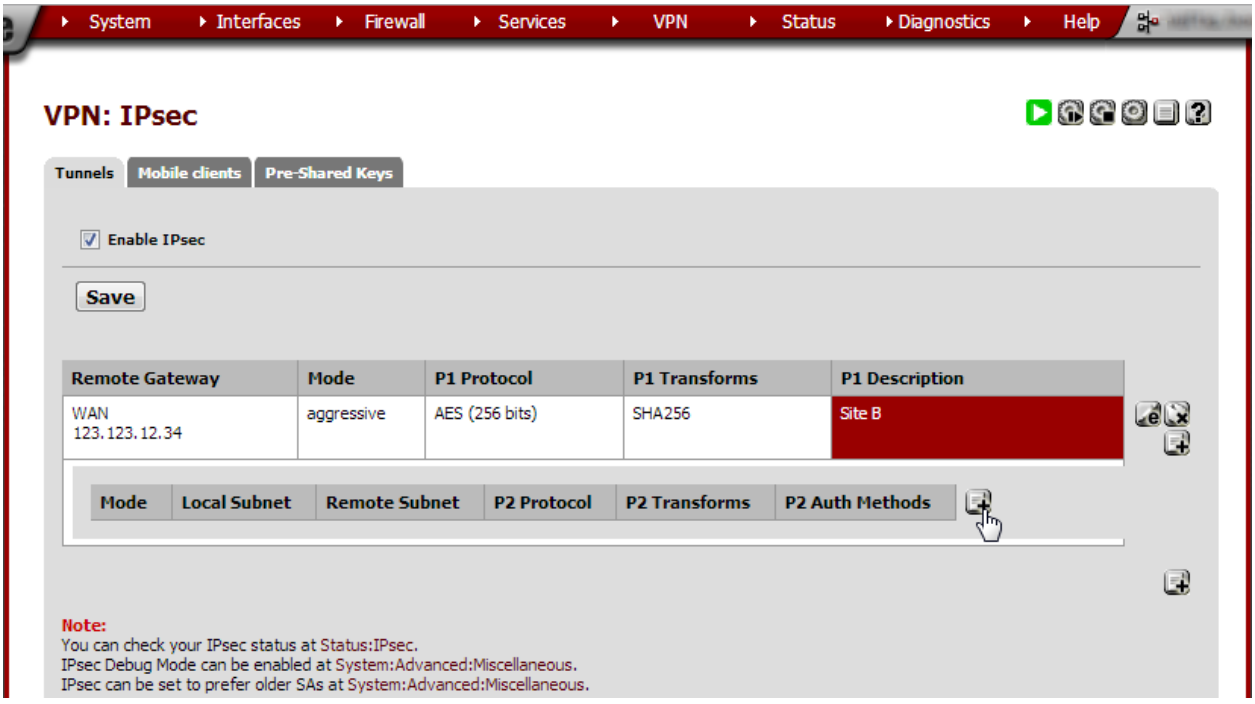
34.34.2 Set up the IPsec tunnel Phase 2

Site A Configuration

Click  under the Phase 1 entry. It will show an overview of all available Phase 2 entries. Since we haven't made any yet none are shown.

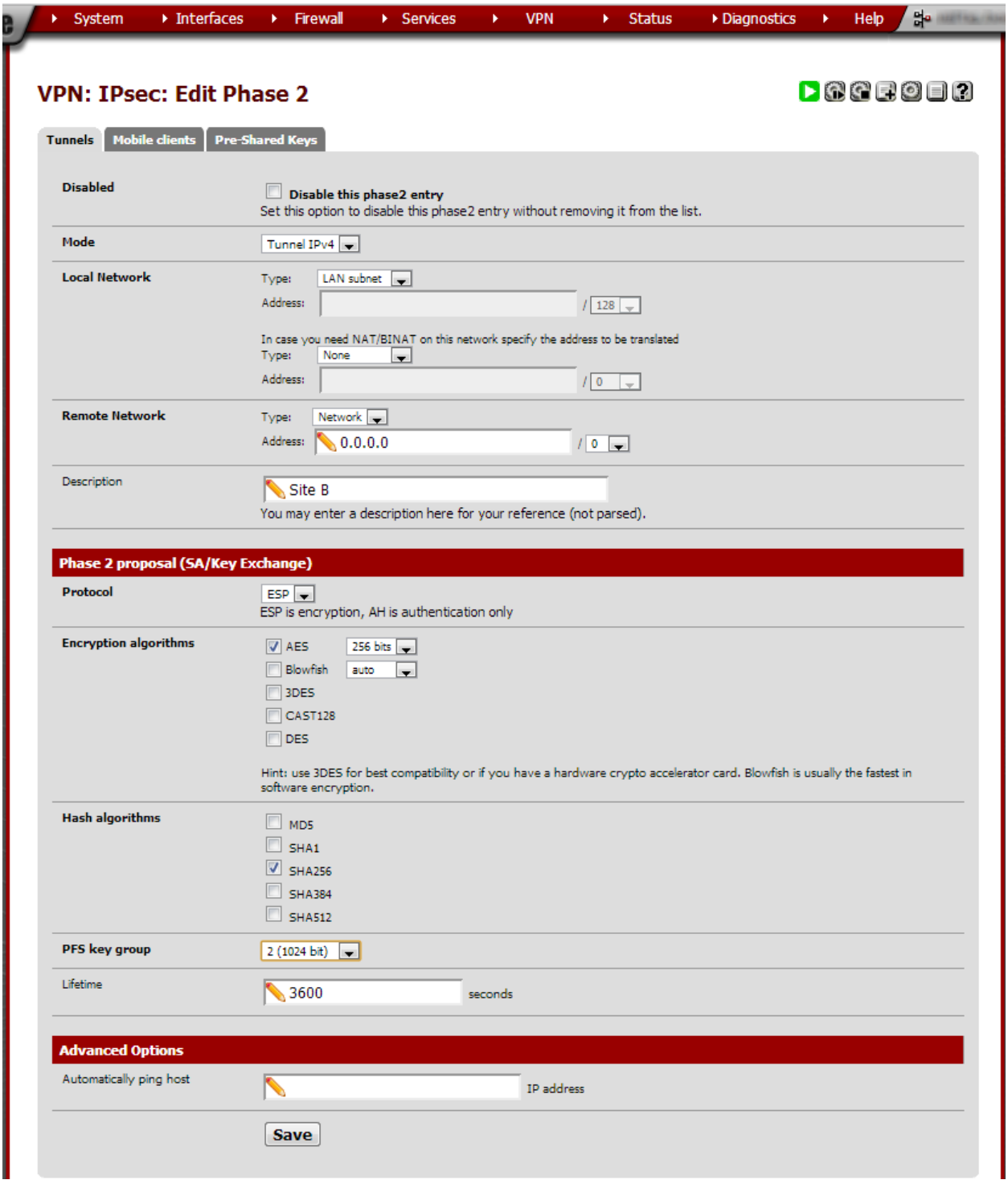


Click  to create a new Phase 2.



Enter these values:

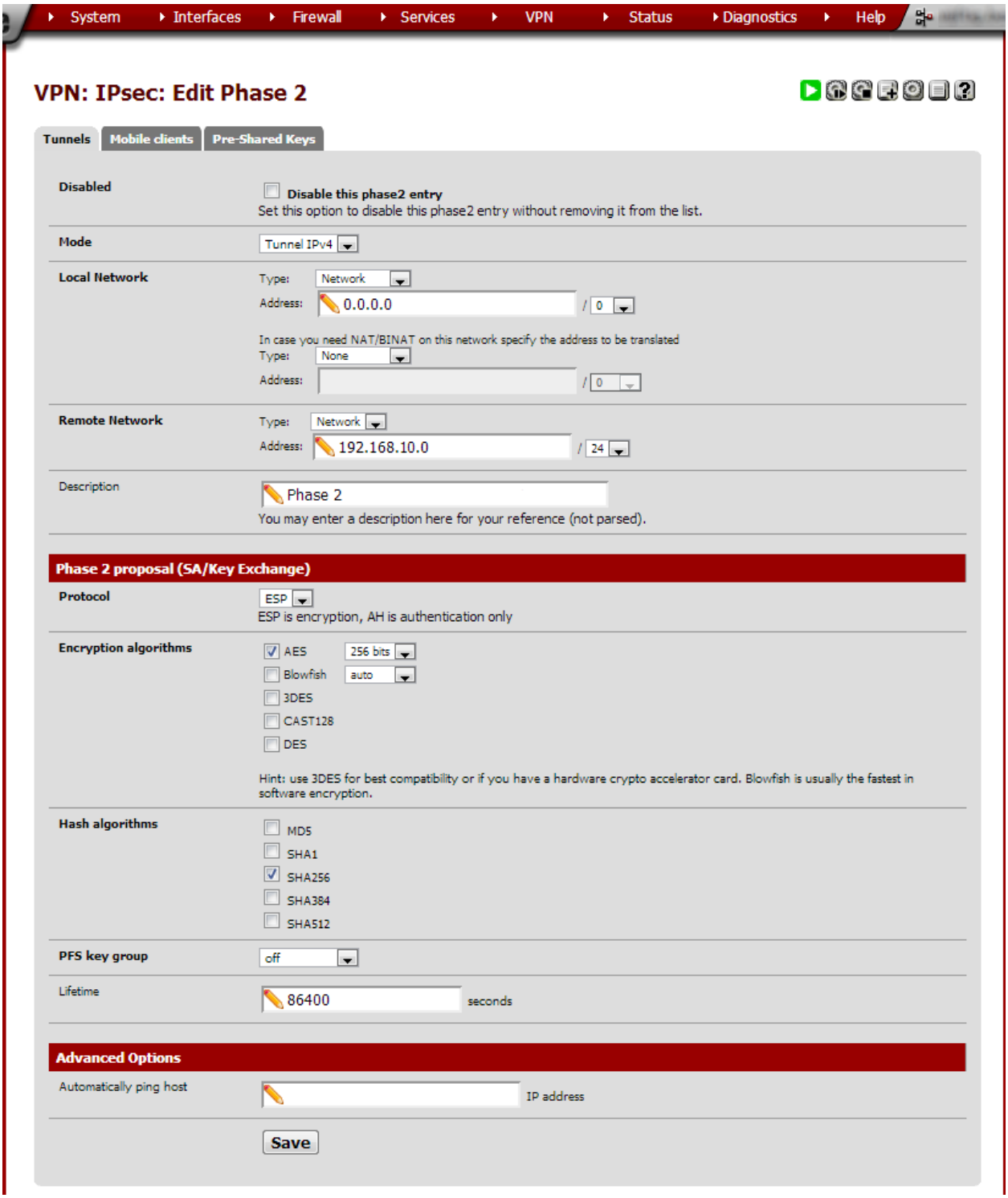
Field	Value	Notes
Mode	Tunnel IPv4	
Local Network	Type: LAN subnet. NAT/BINAT type: None.	
Remote Network	0.0.0.0/0	This tells pfSense to route everything over this interface.
Description	Site B	
Protocol	ESP	
Encryption algorithm	AES 256 bits	
Hash algorithm	SHA256	
PFS key group	2 (1024 bit)	
Lifetime	3600	
Automatically ping host	Enter a hostname or IP address to keep the tunnel alive.	In my experience this is not necessary.



Click **Save** and on the next page click **Apply Changes**.

Site B Configuration

Remote Network, **Type**: Network **Local Network**, **Address**: 0.0.0.0/0 **Remote Network**, **Address**: Site A's LAN subnet Use the same Phase 2 proposal and Advanced options as in Site A.

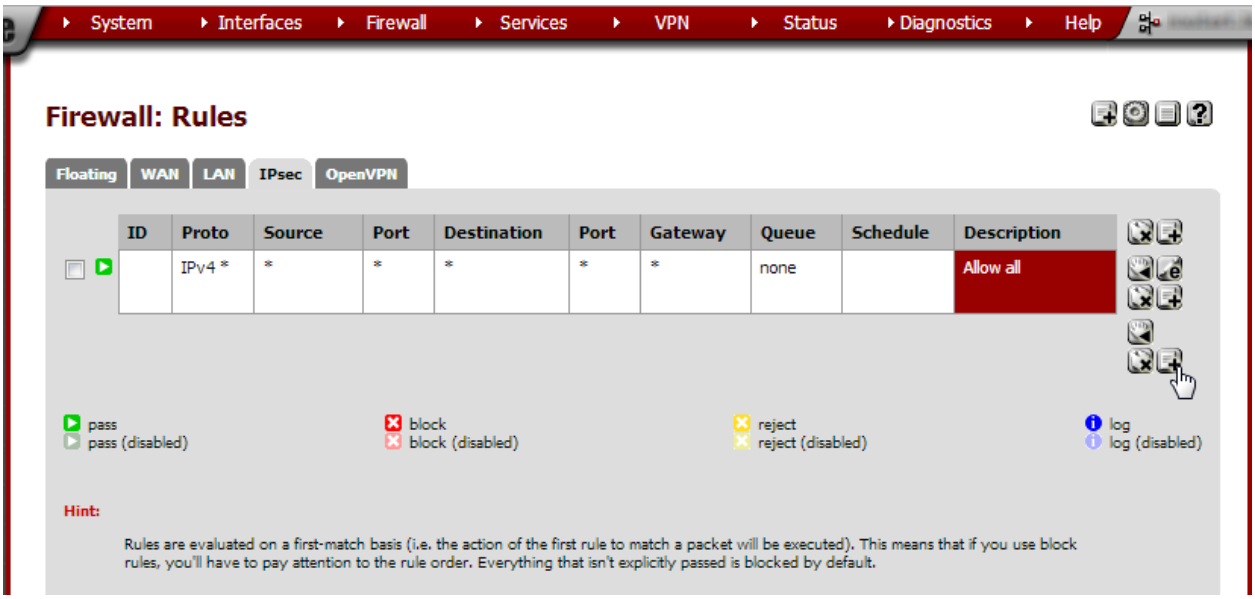


Click **Save** and then **Apply Changes**.

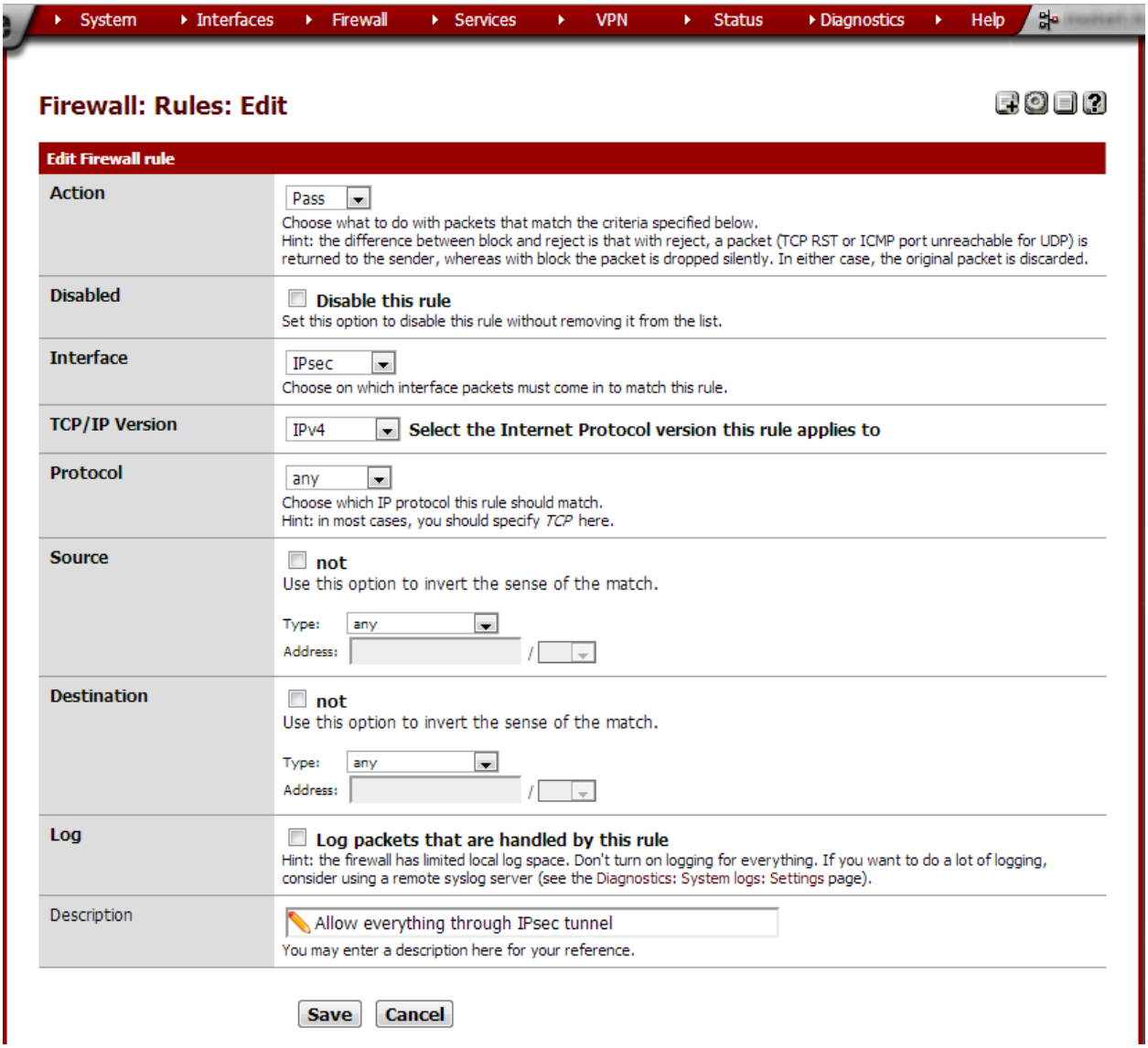
34.34.3 Allow IPsec traffic through the firewall

The tunnel should now be operational however no traffic is allowed through it until a firewall rule is added to pass it. The rule must be added to the routers at both sites.

From the **Firewall** menu, choose **Rules**. Go to the **IPsec tab** and click  .



Set the **Protocol** to **any** and in the Description field type `Allow everything through IPsec tunnel`. Click **Save** and on the next page click **Apply changes**. Do this on both routers.



At this point the tunnel should be up and it should be possible to ping from one side to the other and back. Computers in Site A haven't got an Internet connection however. This is because we still need to configure NAT for the IPsec tunnel.

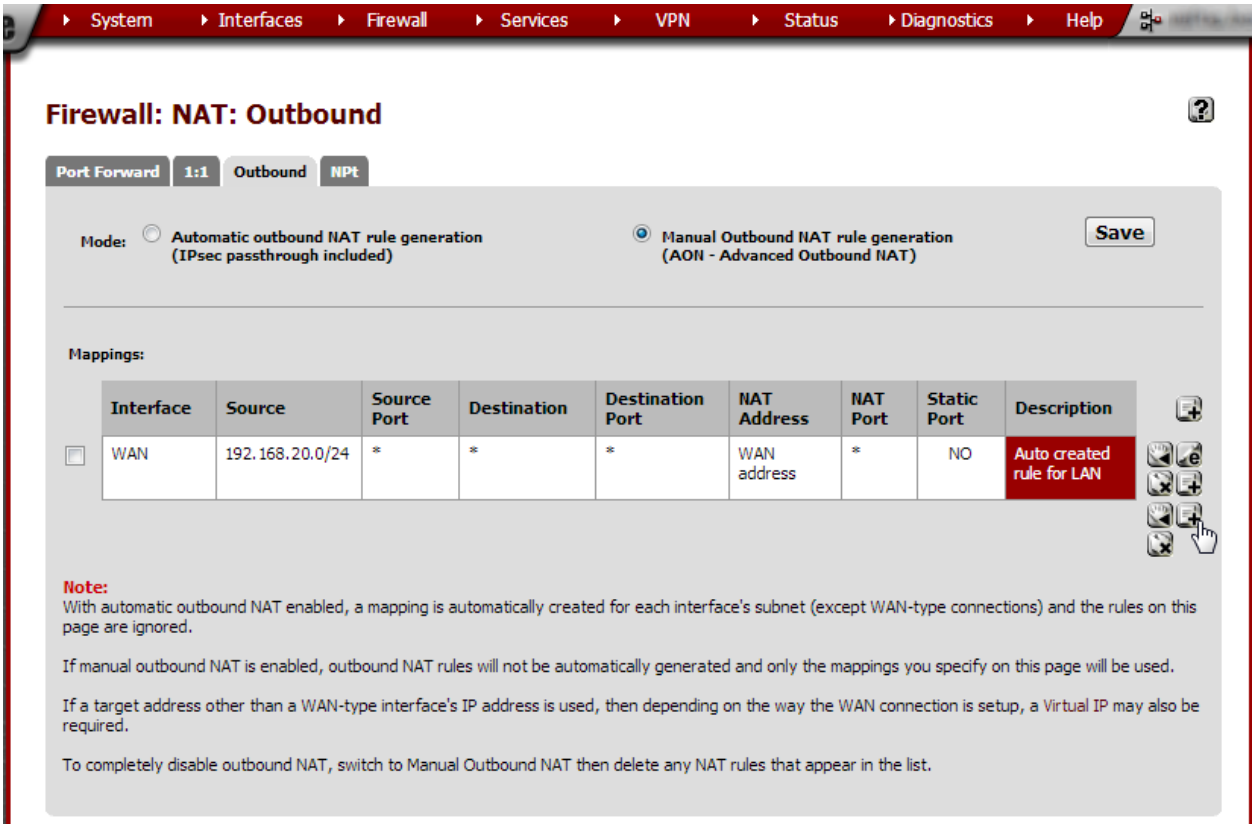
34.34.4 Configure outbound NAT

In the default setup outbound NAT is configured automatically. We need to set it to Manual in order to add Site A's subnet. This configuration step is not required on the router at site A.

Site B Configuration

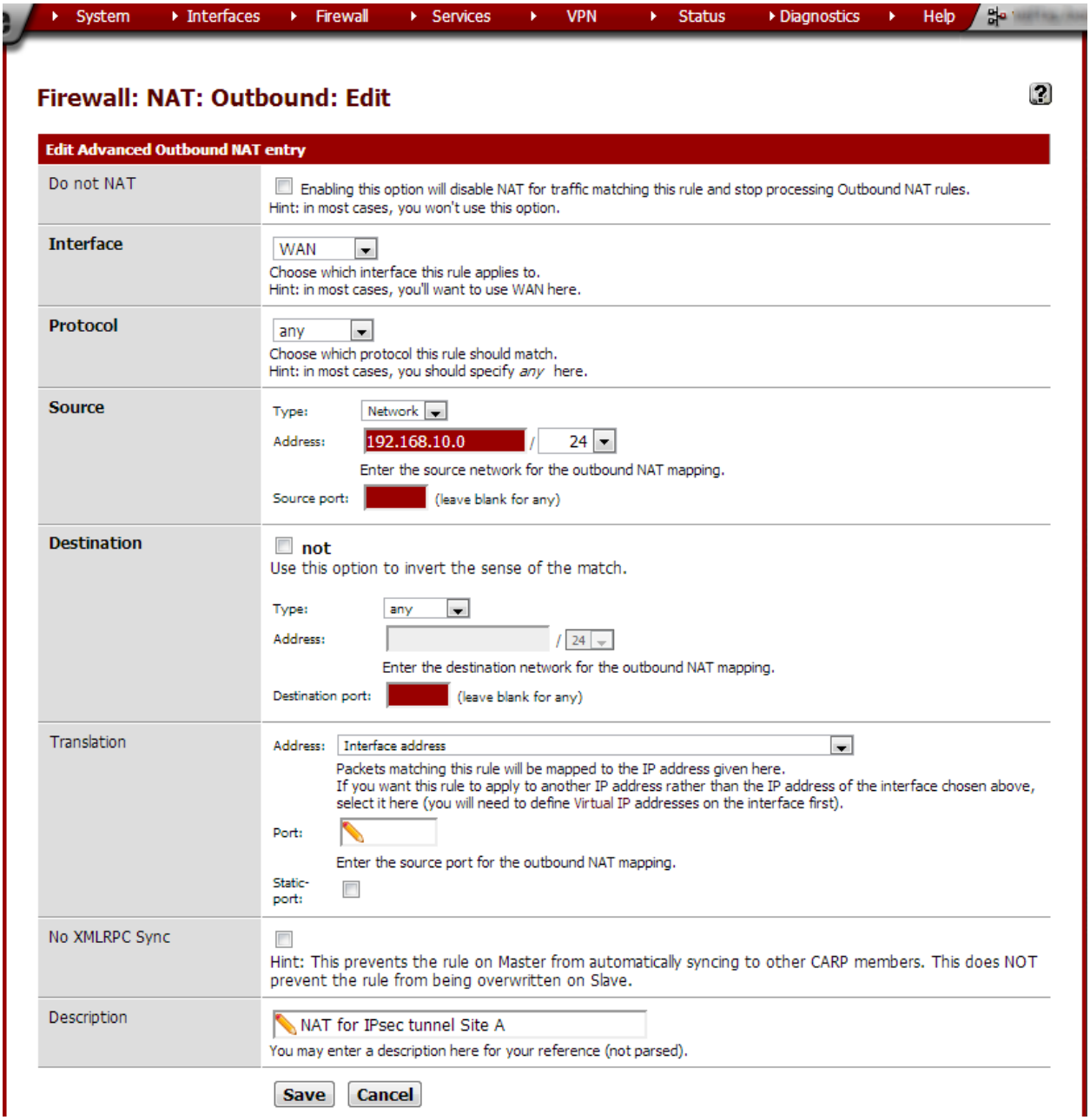
From the **Firewall** menu, choose **NAT** and click the **Outbound tab**. Note that **Mode** is set to **Automatic outbound NAT rule generation**. Select **Manual Outbound NAT rule generation** and click **Save**. On the next page, click **Apply changes**.

Click  to open the **New Mapping page**.

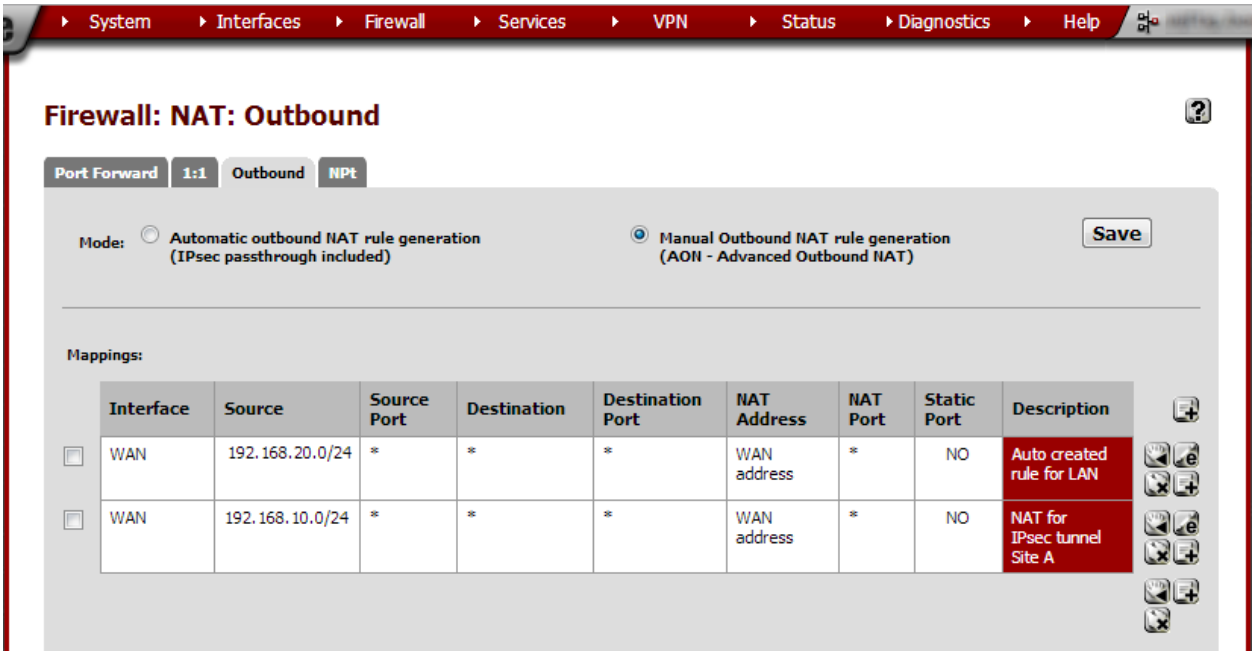


As the **Source Type**, select **Network**. In the **Source Address** field type Site A's subnet: `192.168.10.0/24`.

In the **Description** field, type `NAT for IPsec tunnel Site A`.



Click **Save** and on the next page, click **Apply changes**. The new entry should now be shown in the outbound NAT overview.



At this point Site B will have a working Internet connection through the IPsec tunnel out Site B's Internet provider. Any Internet traffic from Site A will look as if it were coming from Site B (see the diagram at the beginning of this article).

By Vorkbaard, 2013-07-27 - gmail{a}vorkbaard[.]nl, with additional edits.

34.35 IPsec Site-to-Site VPN Example with Certificate Authentication

Using certificate-based authentication for identification of VPN tunnel peers is much stronger than using a simple Pre-Shared Key.

Certificate authentication requires a PKI structure. Depending on the setup, each side may utilize its own CA, or they may share a CA. This example will utilize a different CA on each node, to more closely resemble connecting to third parties. CA and certificate entries can be created and imported in the GUI by the *Certificate Manager*.

34.35.1 Required Information

Endpoint A:

Item	Value
Hostname	office.vpn.example.com
WAN IP Address	`198.51.100.16`

Endpoint B:

Item	Value
Hostname	home.vpn.example.com
WAN IP Address	`198.51.100.17`

34.35.2 Create CA

First, create a Certificate Authority (CA) on each side:

On Endpoint A:

- Navigate to **System > Cert Manager**, **CAs** tab
- Click  **Add**
- Set **Descriptive Name** to `Office VPN CA`
- Set **Method** to *Create an internal Certificate Authority*
- Check **Randomize Serial**
- Set **Common Name** to `office-vpn-ca`
- Leave the rest of the fields at their default values, or adjust to suit local preferences
- Click **Save**
- Click  to export this CA as a file in the browser

On Endpoint B:

- Navigate to **System > Cert Manager**, **CAs** tab
- Click  **Add**
- Set **Descriptive Name** to `Home VPN CA`
- Set **Method** to *Create an internal Certificate Authority*
- Check **Randomize Serial**
- Set **Common Name** to `home-vpn-ca`
- Leave the rest of the fields at their default values, or adjust to suit local preferences
- Click **Save**
- Click  to export this CA as a file in the browser

34.35.3 Import Peer CAs

Next, import the new CA entries into the peer. For example, import the Home CA to the Office side, and vice versa.

Note: This step only requires the certificate data, **not** the key. The key belonging to the CA should not be copied off the firewall where it was created.

On Endpoint A:

- Navigate to **System > Cert Manager**, **CAs** tab
- Click  **Add**

- Set **Descriptive Name** to `Home VPN CA`
- Set **Method** to *Import an existing Certificate Authority*
- Paste the contents of the exported `Home VPN CA.crt` file into **Certificate Data**
- Click **Save**

On Endpoint B:

- Navigate to **System > Cert Manager**, **CAs** tab
- Click  **Add**
- Set **Descriptive Name** to `Office VPN CA`
- Set **Method** to *Import an existing Certificate Authority*
- Paste the contents of the exported `Office VPN CA.crt` file into **Certificate Data**
- Click **Save**

34.35.4 Create Endpoint Certificates

On Endpoint A:

- Navigate to **System > Cert Manager**, **Certificates** tab
- Click  **Add**
- Set **Method** to *Create an internal Certificate*
- Set **Descriptive Name** to `Office VPN Certificate`
- Set **Certificate Authority** to *Office VPN CA*
- Set **Common Name** to `office-vpn-cert`
- Set **Certificate Type** to *User Certificate*
- Set **Alternative Names**, **Type** to *FQDN or Hostname*, **Value** to `office.vpn.example.com`
- Click  **Add**
- Set the new row **Alternative Names**, **Type** to *IP Address*, **Value** to `198.51.100.16`

> **Note:** If the IP address is dynamic, skip this step or use the LAN IP address.

- Leave the rest of the fields at their default values, or adjust to suit local preferences
- Click **Save**

On Endpoint B:

- Navigate to **System > Cert Manager**, **Certificates** tab
- Click  **Add**
- Set **Method** to *Create an internal Certificate*

- Set **Descriptive Name** to `Home VPN Certificate`
- Set **Certificate Authority** to *Home VPN CA*
- Set **Common Name** to `home-vpn-cert`
- Set **Certificate Type** to *User Certificate*
- Set **Alternative Names**, **Type** to *FQDN or Hostname*, **Value** to `home.vpn.example.com`
- Click  **Add**
- Set the new row **Alternative Names**, **Type** to *IP Address*, **Value** to `198.51.100.17`

> **Note:** If the IP address is dynamic, skip this step or use the LAN IP address.

- Leave the rest of the fields at their default values, or adjust to suit local preferences
- Click **Save**

34.35.5 Setup IPsec VPN

On both firewalls, configure the IPsec tunnel as described in *IPsec Site-to-Site VPN Example with Pre-Shared Keys*, with the following exceptions:

Endpoint A:

- Set **Authentication method** to *Mutual Certificate*
- Set **My Identifier** appropriately to *match the certificate* for **this** endpoint
- Set **Peer Identifier** appropriately to *match the certificate* of the **peer**
- Set **My Certificate** to *Office VPN Certificate*
- Set **Peer Certificate Authority** to *Home VPN CA*

Endpoint B:

- Set **Authentication method** to *Mutual Certificate*
- Set **My Identifier** appropriately to *match the certificate* for **this** endpoint
- Set **Peer Identifier** appropriately to *match the certificate* of the **peer**
- Set **My Certificate** to *Home VPN Certificate*
- Set **Peer Certificate Authority** to *Office VPN CA*

34.35.6 Matching Certificate and Identifiers

In order for the IPsec daemon, `strongSwan`, to properly match up a certificate and its keys to a peer, the local and peer identifier must match data in the certificate **exactly**.

> **Warning:** Do not place quotes (single or double) around the identifier values.

There are several ways to accomplish this matching, depending on the environment.

The key factors are:

- The IPsec daemon must be able to confirm that an endpoint matches the expected identifier, which matches a peer to a specific tunnel.
- The IPsec daemon must be able to match that identifier to a certificate and validate its trust, which confirms the identity and authenticates the tunnel peer.

The following identifier types are the best practices to use with certificate authentication:

Distinguished Name This choice can work with fully qualified domain names or short hostnames. If the certificates were created as specified in *Create Endpoint Certificates*, use the full hostname such as `office.vpn.example.com` or `home.vpn.example.com`. This is the easiest choice and most likely to succeed, assuming the SAN value is present in the certificate.

Modern certificates typically include the certificate CN as a SAN entry, so the CN may also be used if it resembles a hostname (e.g. `office-vpn-cert`). Check the certificate properties to ensure it is present as an FQDN SAN entry. This will not work if the CN contains spaces or other characters not compatible with hostnames.

ASN.1 Distinguished Name The full ASN.1 Distinguished Name of the certificate. This is similar to the certificate subject but has stricter rules about its order.

This can be formatted in several ways so long as it matches the data in the certificate exactly, for example:

- `/CN=host.example.com/C=US/ST=Texas/L=Austin/O=Example Co`
- `CN=host.example.com, C=US, ST=Texas, L=Austin, O=Example Co`
- `CN = host.example.com, C = US, ST = Texas, L = Austin, O = Example Co`

Note: The type, number, and order of fields will vary depending on how the certificate was made.

To find this string, inspect the certificate in one of the following ways:

- From the **Certificate Manager**, **Certificates** tab, find the entry and click the  icon to expand the certificate details. In the details, copy the contents of the **DN:** field exactly.

```
DN: /CN=host.example.com/C=US/ST=Texas/L=Austin/O=Example Co
```

- Use OpenSSL on a copy of the certificate contents and look for the **Subject** contents:

```
$ openssl x509 -text -noout -in mycert.crt | grep Subject:
        Subject: CN = host.example.com, C = US, ST = Texas, L =␣
↪Austin, O = Example Co
```

- If the certificate is configured in IPsec already, look at how `strongSwan` reports the certificate subject:

```
$ swanctl --list-certs | grep subject
  subject:  "CN=host.example.com, C=US, ST=Texas, L=Austin, O=Example␣
↪Co"
```

Warning: When copying these values, remember that they must be entered exactly as shown but **without** any single or double quotes around the string. Only include the DN contents and not any headers or labels such as `DN:` or `Subject:`.

My IP Address / Peer IP Address These choices are viable if all of the following items are true:

- Both endpoints have static IP addresses
- These static IP addresses are used as the **Remote Gateway** address on each side of the IPsec tunnel
- The static IP address of an endpoint is present in its certificate as a SAN

IP Address Similar to the **My IP Address / Peer IP Address** case above, but instead of using endpoint static IP addresses, uses a pre-determined local addresses instead. This could be the LAN IP address or another agreed upon address which does not change. This value does not need to match the **Remote Gateway** address in this case.

- The value must be present as an IP address type SAN in the certificate

In most cases, this is not ideal, and the hostname is easier to use instead.

34.35.7 Troubleshooting

If the IPsec daemon cannot match an identifier to a known certificate, the following error is logged on one or both of the peers:

```
charon[5319]: 08[IKE] <con100000|1> no trusted RSA public key found for '<identifier>'
```

In that case:

- Check over all of the identifier data again to ensure that the values **exactly** match an appropriate certificate field (DN, SAN, etc.)
- If using an ASN.1 DN, ensure the order of DN/subject components **exactly** matches the order reported by the DN field in the Certificate Manager, `strongSwan`, or `openssl`
- Ensure there are no single or double quotes around the identifier value in the GUI
- Ensure the correct **Peer Certificate Authority** is imported and selected

When attempting a connection, locate the correct `con` identifier in strongswan by comparing the tunnel settings against the output of the following command in an SSH or console shell:

```
# swanctl --list-conns
```

Then attempt to initiate the tunnel with:

```
# swanctl --initiate --ike conXXXXXXX
```

When initiating a tunnel in this way, `swanctl` will output only the relevant logs to the terminal. This is much easier than attempting to follow the log file contents in other ways.

Attempt to initiate the tunnel in both directions and compare output.

34.36 Configuring IPv6 Through A Tunnel Broker Service

A location that doesn't have access to native IPv6 connectivity may obtain it using a tunnel broker service such as Hurricane Electric. A core site with IPv6 can deliver IPv6 connectivity to a remote site by using a VPN or GIF tunnel.

This section provides the process for connecting pfSense® with Hurricane Electric (Often abbreviated to HE.net or HE) for IPv6 transit. Using HE.net is simple and easy. It allows for multi-tunnel setup, each with a transport /64 and a routed /64. Also included is a routed /48 to be used with one the tunnels. It's a great way to get a lot of routed IPv6 space to experiment with and learn, all for free.

34.36.1 Sign Up for Service

Before a tunnel can be created, ICMP echo requests must be allowed to the WAN. A rule to pass ICMP echo requests from a source of *any* is a good temporary measure. Once the tunnel endpoint for HE.net has been chosen, the rule can be made more specific.

To get started on HE.net, sign up at www.tunnelbroker.net. The /64 networks are allocated after registering and selecting a regional IPv6 tunnel server. A summary of the tunnel configuration can be viewed on HE.net's website as seen in Figure *HE.net Tunnel Config Summary*. It contains important information such as the user's **Tunnel ID**, **Server IPv4 Address** (IP address of the tunnel server), **Client IPv4 Address** (the firewall's external IP address), the **Server** and **Client IPv6 Addresses** (representing the IPv6 addresses inside the tunnel), and the **Routed IPv6 Prefixes**.

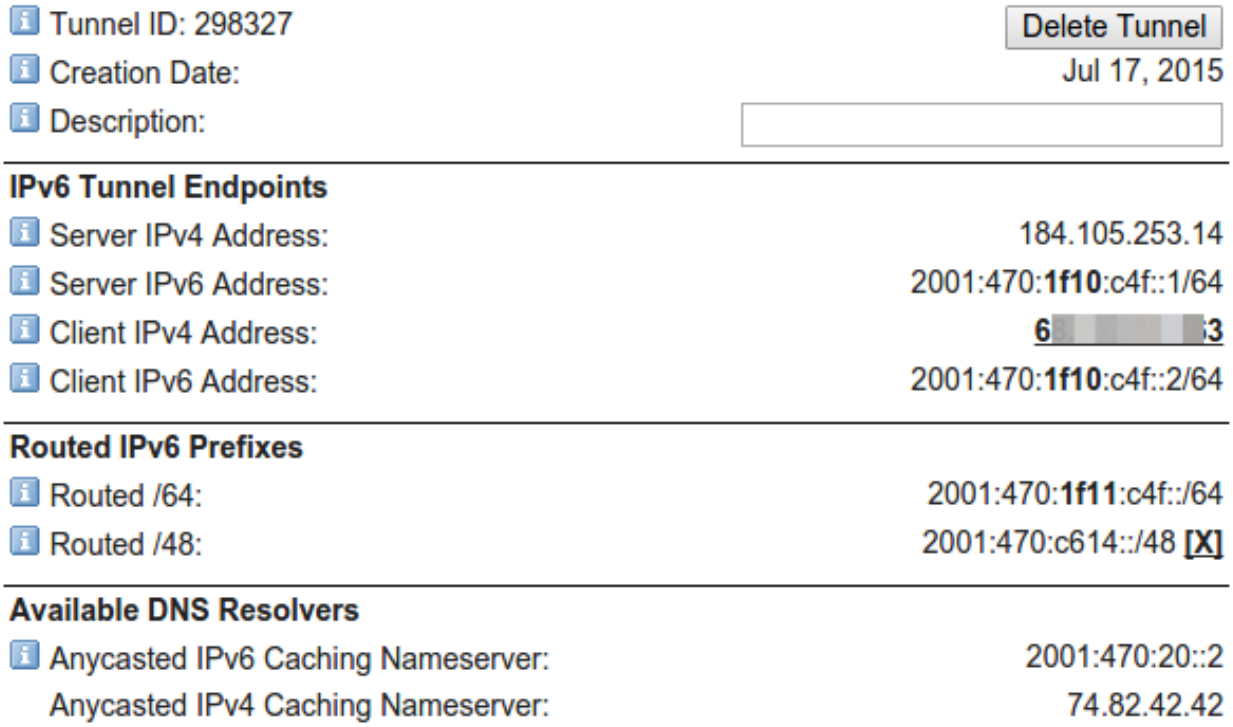


Fig. 60: HE.net Tunnel Config Summary

The **Advanced** tab on the tunnel broker site has two additional notable options: An MTU Slider and an Update Key for updating the tunnel address. If the WAN used for terminating the GIF tunnel is PPPoE or another WAN type with a low MTU, move the slider down as needed. For example, a common MTU for PPPoE lines with a tunnel broker would be 1452. If the WAN has a dynamic IP address, note the **Update Key** for later use in this section.

Once the initial setup for the tunnel service is complete, configure pfSense to use the tunnel.

34.36.2 Allow IPv6 Traffic

On new installations of pfSense after 2.1, IPv6 traffic is allowed by default. If the configuration on the firewall has been upgraded from older versions, then IPv6 would still be blocked. To enable IPv6 traffic, perform the following:

- Navigate to **System > Advanced** on the **Networking** tab
- Check **Allow IPv6** if not already checked
- Click **Save**

34.36.3 Allow ICMP

ICMP echo requests must be allowed on the WAN address that is terminating the tunnel to ensure that it is online and reachable. If ICMP is blocked, the tunnel broker may refuse to setup the tunnel to the IPv4 address. Edit the ICMP rule made earlier in this section, or create a new rule to allow ICMP echo requests. Set the source IP address of the **Server IPv4 Address** in the tunnel configuration as shown in Figure *Example ICMP Rule* to ensure connectivity.



Fig. 61: Example ICMP Rule

34.36.4 Create and Assign the GIF Interface

Next, create the interface for the GIF tunnel in pfSense. Complete the fields with the corresponding information from the tunnel broker configuration summary.

- Navigate to **Interfaces > Assignments** on the **GIF** tab.
- Click **Add** to add a new entry.
- Set the **Parent Interface** to the WAN where the tunnel terminates. This would be the WAN which has the **Client IPv4 Address** on the tunnel broker.
- Set the **GIF Remote Address** in pfSense to the **Server IPv4 Address** on the summary.
- Set the **GIF Tunnel Local Address** in pfSense to the **Client IPv6 Address** on the summary.
- Set the **GIF Tunnel Remote Address** in pfSense to the **Server IPv6 Address** on the summary, along the with prefix length (typically */ 64*).
- Leave remaining options blank or unchecked.
- Enter a **Description**.
- Click **Save**.

See Figure *Example GIF Tunnel*.

If this tunnel is being configured on a WAN with a dynamic IP, see *Updating the Tunnel Endpoint* for information on how to keep the tunnel's endpoint IP updated with HE.net.

Once the GIF tunnel has been created, it must be assigned:

- Navigate to **Interfaces > Assignments**, **Interface Assignments** tab.
- Select the newly created GIF under **Available Network Ports**.

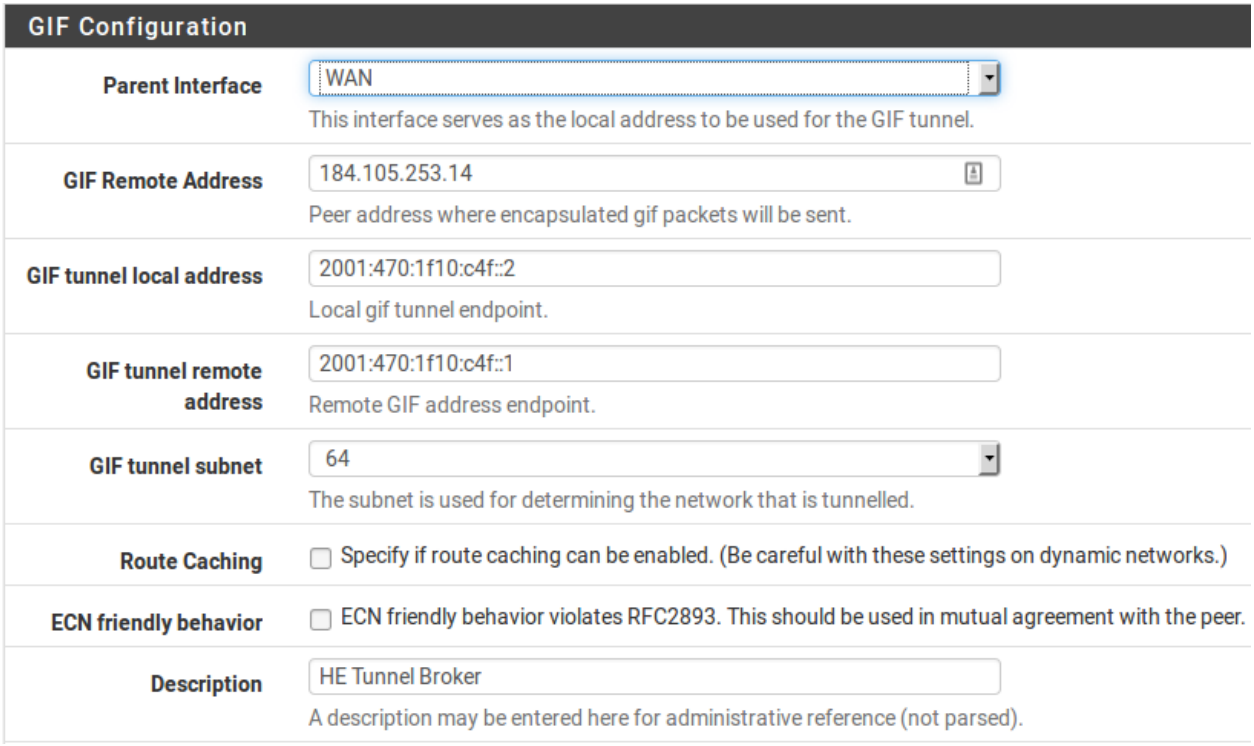


Fig. 62: Example GIF Tunnel

- Click  **Add** to add it as a new interface.

34.36.5 Configure the New OPT Interface

The new interface is now accessible under **Interfaces > OPTx**, where *x* depends on the number assigned to the interface.

- Navigate to the new interface configuration page. (**Interfaces > OPTx**)
- Check **Enable Interface**.
- Enter a name for the interface in the **Description** field, for example *WANv6*.
- Leave **IPv6 Configuration Type** as *None*.
- Click **Save**
- Click **Apply Changes**.

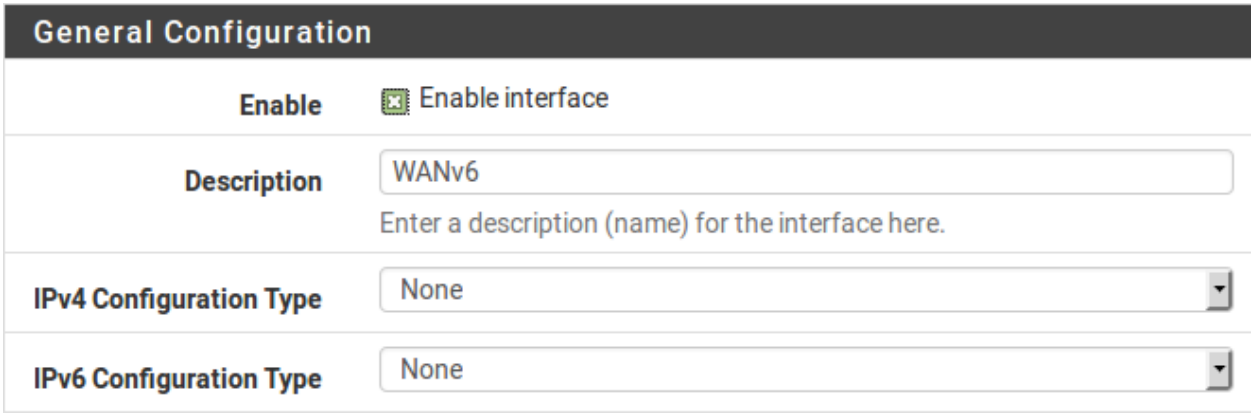


Fig. 63: Example Tunnel Interface

34.36.6 Setup the IPv6 Gateway

When the interface is configured as listed above, a dynamic IPv6 gateway is added automatically, but it is not yet marked as default.

- Navigate to **System > Routing**
- Edit the dynamic IPv6 gateway with the same name as the IPv6 WAN created above.
- Check **Default Gateway**.
- Click **Save**.
- Click **Apply Changes**.

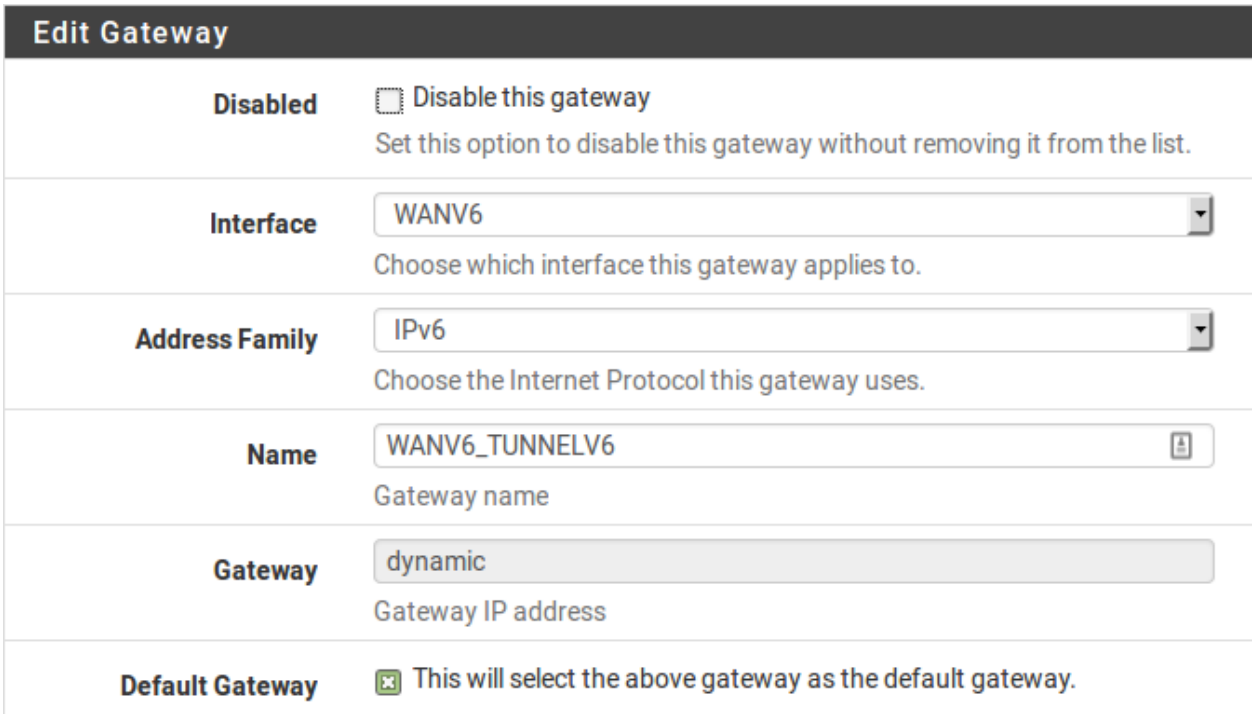


Fig. 64: Example Tunnel Gateway

Navigate to **Status > Gateways** to view the gateway status. The gateway will show as "Online" if the configuration is successful, as seen in Figure *Example Tunnel Gateway Status*.

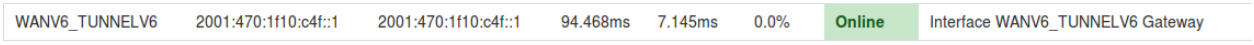


Fig. 65: Example Tunnel Gateway Status

34.36.7 Setup IPv6 DNS

The DNS servers likely answer DNS queries with AAAA results already. Entering the DNS servers supplied by the tunnel broker service under **System > General Setup** is recommended. Enter at least one IPv6 DNS server or use Google's public IPv6 DNS servers at 2001:4860:4860::8888 and 2001:4860:4860::8844. If the DNS Resolver is used in non-forwarding mode, it will talk to IPv6 root servers automatically once IPv6 connectivity is functional.

34.36.8 Setup LAN for IPv6

Once the tunnel is configured and online, the firewall itself has IPv6 connectivity. To ensure clients can access the internet on IPV6, the LAN must be configured also.

One method is to set LAN as dual stack IPv4 and IPv6.

- Navigate to **Interfaces > LAN**
- Select **IPv6 Configuration Type** as **Static IPv6**
- Enter an IPv6 address from the **Routed /64** in the tunnel broker configuration with a prefix length of *64*. For example, * *2001:db8:1111:2222::1* for the LAN IPv6 address if the **Routed /64** is *2001:db8:1111:2222::/64*.
- Click **Save**
- Click **Apply Changes**

A /64 from within the Routed /48 is another available option.

34.36.9 Setup DHCPv6 and/or Router Advertisements

To assign IPv6 addresses to clients automatically, setup Router Advertisements and/or DHCPv6. This is covered in detail in *IPv6 Router Advertisements*.

A brief overview is as follows:

- Navigate to **Services > DHCPv6 Server/RA**
- Check **Enable**
- Enter a range of IPv6 IP addresses inside the new LAN IPv6 subnet
- Click **Save**.
- Switch to the **Router Advertisements** tab
- Set the **Mode** to *Managed* (DHCPv6 only) or *Assisted* (DHCPv6+SLAAC)
- Click **Save**.

Modes are described in greater detail at *Router Advertisements (Or: "Where is the DHCPv6 gateway option")*.

To assign IPv6 addresses to LAN systems manually, use the firewall's LAN IPv6 address as the gateway with a proper matching prefix length, and pick addresses from within the LAN subnet.

34.36.10 Add Firewall Rules

Once LAN addresses have been assigned, add firewall rules to allow the IPv6 traffic to flow.

- Navigate to **Firewall > Rules**, **LAN** tab.
- Check the list for an existing IPv6 rule. If a rule to pass IPv6 traffic already exists, then no additional action is necessary.
- Click  **Add** to add a new rule to the bottom of the list
- Set the **TCP/IP Version** to *IPv6*
- Enter the LAN IPv6 subnet as the **Source**
- Pick a **Destination** of *Any*.
- Click **Save**
- Click **Apply Changes**

For IPv6-enabled servers on the LAN with public services, add firewall rules on the tab for the IPv6 WAN (the assigned GIF interface) to allow IPv6 traffic to reach the servers on required ports.

34.36.11 Try It!

Once firewall rules are in place, check for IPv6 connectivity. A good site to test with is test-ipv6.com. An example of the output results of a successful configuration from a client on LAN is shown here Figure *IPv6 Test Results*.

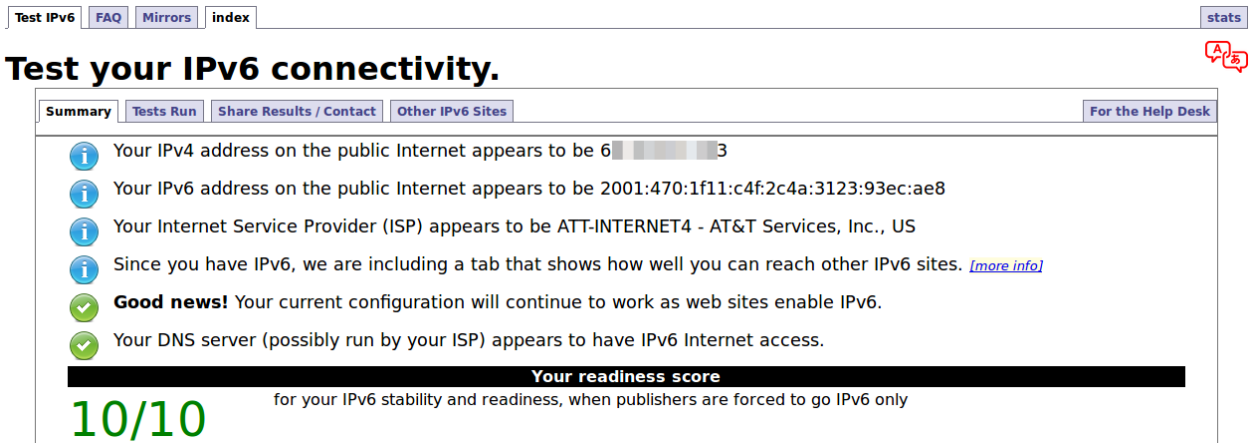


Fig. 66: IPv6 Test Results

34.36.12 Updating the Tunnel Endpoint

For a dynamic WAN, such as DHCP or PPPoE, HE.net can still be used as a tunnel broker. pfSense includes a DynDNS type that will update the tunnel endpoint IP address whenever the WAN interface IP changes.

If DynDNS is desired, it may be configured as follows:

- Navigate to **Services > DynDNS**
- Click  **Add** to add a new entry.
- Set the **Service Type** to be *HE.net Tunnelbroker*.
- Select *WAN* as the **Interface to Monitor**.
- Enter the **Tunnel ID** from the tunnel broker configuration into the **Hostname** field.
- Enter the **Username** for the tunnel broker site.
- Enter either the **Password** or **Update Key** for the tunnel broker site into the **Password** field.
- Enter a **Description**.
- Click **Save and Force Update**.

If and when the WAN IP address changes, pfSense will automatically update the tunnel broker.

34.37 L2TP/IPsec Remote Access VPN Configuration Example

On current versions of pfSense® software, L2TP/IPsec may be configured for mobile clients, though it is not a configuration we recommend.

> **Warning:** Users have reported issues with Windows L2TP/IPsec clients behind NAT. If the clients will be behind NAT, Windows clients will most likely not function. Consider an IKEv2 implementation instead.

As warned at the start of the chapter, the Windows client, among others, and the strongSwan IPsec daemon are not always compatible, leading to failure in many cases. We strongly recommend using another solution such as IKEv2 instead of L2TP/IPsec.

See also:

IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2 contains a walkthrough for configuring IKEv2.

Before configuring the IPsec portion, setup the L2TP server as described in *L2TP Server Configuration* and add users, firewall rules, etc, as covered there.

34.37.1 Setup IPsec

These settings have been tested and found to work with some clients, but other similar settings may function as well. Feel free to try other encryption algorithms, hashes, etc.

Mobile Clients Tab

- Navigate to **VPN > IPsec**, **Mobile Clients** tab in the pfSense WebGUI
- Check **Enable IPsec Mobile Client Support**
- Set **User Authentication** to *Local Database* (Not used, but the option must have something selected)
- Uncheck **Provide a virtual IP address to clients**
- Uncheck **Provide a list of accessible networks to clients**
- Click **Save**

Phase 1

- Click the **Create Phase1** button at the top if it appears, or edit the existing Mobile IPsec Phase 1
  - If there is no Phase 1, and the **Create Phase1** button does not appear, navigate back to the **Mobile Clients** tab and click it there.
- Set **Key Exchange version** to *v1* or *Auto*
- Enter an appropriate **Description**
- Set **Authentication method** to *Mutual PSK*
- Set **Negotiation Mode** to *Main*
- Set **My Identifier** to *My IP address*
- Set **Encryption algorithm** to *AES 256*
- Set **Hash algorithm** to *SHA1*
- Set **DH key group** to *14 (2048 bit)*

Note: iOS and other platforms may work with a **DH key group** of *2* instead.

- Set **Lifetime** to `28800`
- Uncheck **Disable Rekey**
- Set **NAT Traversal** to *Auto*
- Check **Enable DPD**, set for *10* seconds and *5* retries
- Click **Save**

Phase 2

- Click  **Show Phase 2 Entries** to show the Mobile IPsec Phase 2 list
- Click  **Add P2** to add a new Phase 2 entry if one does not exist, or click  to edit an existing entry
- Set **Mode** to **Transport**
- Enter an appropriate **Description**
- Set **Protocol** to *ESP*
- Set **Encryption algorithms** to ONLY *AES 128*
- Set **Hash algorithms** to ONLY *SHA1*
- Set **PFS Key Group** to *off*
- Set **Lifetime** to `3600`
- Click **Save**

Pre-Shared Key

The Pre-Shared Key for the connection, which is common for all clients, must be configured in a special way.

- Navigate to **VPN > IPsec**, **Pre-Shared Keys** tab on pfSense
- Click  **Add** to add a new PSK
- Set the **Identifier** to `allusers`

> **Note:** The `allusers` name is a special keyword used by pfSense to configure a wildcard PSK, which is necessary for L2TP/IPsec to function. Do not use any other **Identifier** for this PSK!

- Set **Secret Type** to *PSK*
- Enter a **Pre-Shared Key**, such as `aaabbbccc` – ideally one a lot longer, more random, and secure!
- Click **Save**
- Click **Apply Changes**

34.37.2 IPsec Firewall Rules

Firewall rules are necessary to pass traffic from the client host over IPsec to establish the L2TP tunnel, and inside L2TP to pass the actual tunneled VPN traffic to systems across the VPN. Adding the L2TP rules was covered in the previous section. To add IPsec rules:

- Navigate to **Firewall > Rules**, **IPsec** tab
- Review the current rules. If there is an "allow all" style rule, then there is no need to add another. Continue to the next task.
- Click  **Add** to add a new rule to the top of the list

- Set the **Protocol** to *any*
- Set the **Source** and **Destination** to *any*

> **Note:** This does not have to pass all traffic, but must at least pass L2TP (UDP port `1701`) to the WAN IP address of the firewall

- Click **Save**
- Click **Apply Changes**

34.37.3 DNS Configuration

If DNS servers are supplied to the clients and the Unbound **DNS Resolver** is used, then the subnet chosen for the L2TP clients must be added to its access list.

- Navigate to **Services > DNS Resolver**, **Access Lists** tab
- Click  **Add** to add a new access list
- Enter an **Access List Name**, such as *VPN Users*
- Set **Action** to *Allow*
- Click  **Add Network** under **Networks** to add a new network
- Enter the VPN client subnet into the **Network** box, e.g. `10.3.177.128`
- Choose the proper **CIDR**, e.g. `25`
- Click **Save**
- Click **Apply Changes**

34.37.4 Client Setup

When configuring clients, there are a few points to look for:

- Ensure that the client operating system configuration is set to connect to the proper external address for the VPN.
- It may be necessary to force the VPN type to **L2TP/IPsec** on the client if it has an automatic mode.
- The client authentication type must match what is configured on the L2TP server (e.g. *CHAP*)

34.38 Connecting to L2TP/IPsec from Android

The L2TP/IPsec client on Android has the ability to set a custom identifier, which allows L2TP/IPsec to function with the pfSense® server using Pre-Shared Keys. Clients on other operating systems do not allow for this, which makes them incompatible with current versions of pfSense software.

34.38.1 IPsec Setup

The setup is similar to a standard *IPsec Remote Access VPN Example Using IKEv1 with Xauth* setup except that xauth is not used, but rather "**Mutual PSK**", and Phase 2 uses **Transport** mode rather than Tunnel.

Pre-Shared Keys

After the tunnel has been configured, click to the "Pre-Shared Keys" tab in the IPsec settings, and add IPsec keys. A single group key may be used if desired, or make many keys for different users.

That's it for IPsec!

34.38.2 L2TP Setup

To setup L2TP navigate to VPN > L2TP

- Select **Enable L2TP Server**
- **Interface** is **WAN** (or the same chosen for IPsec)
- **Server Address** is an **unused IP address** in a new subnet. It **MUST NOT** overlap any IP in use on the firewall, e.g x.x.x.2
- **Remote Address Range** is the starting IP of the clients, e.g. x.x.x.128
- **Subnet netmask** is the netmask for the client connection, the server IP should be included in this subnet, e.g. /24
- **Secret** should be **left blank**, it does not appear to work, at least with the Android version tested.
- **Encryption Type**: CHAP is recommended
- **L2TP DNS Servers**: The firewall's actual LAN IP, or another internal DNS server
- RADIUS settings - if needed, use them, otherwise leave them alone
- Save
- Flip to the Users tab and add L2TP user accounts and passwords there
- Now go to **Firewall > Rules** on the **L2TP VPN** tab, and add a firewall rule to pass traffic, e.g from any to any or much more restrictive if preferred.

34.38.3 Android Client Setup

On the phone/tablet/device:

- Go to the system settings and VPN settings (varies by device and specific Android version
- Tap **Add VPN Profile**
- Enter a name
- For **Type**, tap *L2TP/IPsec PSK*
- **Server Address**: The WAN IP of the pfSense router (or the IP of the interface chosen for IPsec and L2TP)
- **L2TP Secret**: **Left blank**
- **IPsec Identifier**: Enter the identifier for the PSK entered above, either a per-user or common identifier
- **IPsec Pre-Shared Key**: The PSK that goes with the identifier for this user/group

- The advanced options may be used to control which networks will attempt to use the VPN, or specify custom DNS server and domains for this client.
- Tap **Save**
- From the VPN list, tap the newly created VPN entry
- Enter the **username** and **password** from the **L2TP Users** tab entered above
- Check **Save account information** to save the VPN credentials (not recommended!),
- Tap **Connect**

The connection should then connect and function. If it does not work, check the IPsec logs and the **Status > System Logs**, **VPN**, **L2TP Raw log** to see more specific errors.

34.38.4 Other Thoughts

In theory, Mutual RSA should also work, but so far it has not succeeded in testing. In RSA mode, Phase 1 requires main mode, but otherwise should be OK.

34.39 Migrating an Assigned LAN to LAGG

Only unassigned physical ports can be added to a LAGG, so to move an assigned LAN interface to a LAGG requires some shuffling around.

In this example, the LAN of an APU (*re2*) will be moved into a LAGG with the OPT1 port (re0).

34.39.1 Warnings/Precautions

It is best to perform this change from an interface that is not involved, such as a DMZ, OPT port, perhaps WAN or VPN. Though if the switch is properly configured there should be no loss of connectivity, at least with LACP. Other modes may vary.

34.39.2 Prerequisites/Assumptions

The switch must be properly configured to accommodate the LAGG. This typically means configuring an LACP group and setting ports to use that group. The NICs involved, in this example re0 and re2, should be connected to properly configured ports on the switch before starting.

34.39.3 Migrate LAN to a LAGG

1. Ensure the second NIC for the LAGG is not assigned (e.g. *re0* mapped to *OPT1*)
    1. In the pfSense® webGUI, check **Interfaces > Assignments** and remove its entry if present
2. Create a new LAGG including only the second NIC
    1. Navigate to **Interfaces > Assignments** on the **LAGG** tab
    2. Click  to create a new LAGG
    3. Click to select the NIC to use with this LAGG (*re0*)

4. Select the proper LAGG protocol, such as *LACP*
5. Enter a description
6. Click **Save**

3. Navigate to **Interfaces > Assignments**, change the assignment of LAN to the newly created LAGG interface (*LAGG0*)
4. Click **Save**
5. Navigate back to the **LAGG** tab
6. Click  to edit the LAGG
7. Ctrl-click to add in the port that was formerly assigned to LAN (*re2*) so that both NICs are selected
8. Click **Save**

Now both *re0* and *re2* are members of a LAGG and that LAGG is the LAN interface with all of the existing configuration in place.

34.39.4 VLANS

If any VLANs were in use directly on the interfaces involved, migrate them as follows:

- Add new VLAN tags using the LAGG interface as the parent (**Interfaces > Assignments**, **VLAN** tab)
- Fix the assignments to use the LAGG versions of the tags (**Interfaces > Assignments**)
- Remove the old tags from the physical interface(s) (**Interfaces > Assignments**, **VLAN** tab)

Do not edit the existing tags and change the parent interface, it will cause problems with the interface assignments. Always create new tags, switch the assignments, then remove the old tags.

34.40 Accessing a CPE/Modem from Inside the Firewall

Most end-user Customer Premise Equipment (CPE) devices like cable or DSL modems have a web interfaces on a private IP address. Since these sit outside the firewall and do not typically have a public IP address, accessing them isn't as straight forward as it might seem. The firewall is typically assigned a public IP, and sends all outbound traffic upstream to the ISP. The ISP won't route the private subnet back to the modem, leaving it unreachable. This page describes the work around needed to access the management interface on the modem from the inside of the network.

Note: The CPE management IP address **must** be on a different IP subnet than the internal network. If it is not, attempts to connect to it will never go to the firewall to be routed out to the modem, as hosts on the internal network would try to connect to it on the local network and fail.

34.40.1 Configure a new Interface

A PPPoE WAN is actually assigned to a virtual PPPoE adapter, not the physical port.

Under **Interfaces > Assignments**, create a new *OPT* interface, and assign it to the physical network card that is on WAN. For example, if the WAN on the assignment page is "*PPPOE0(re2)*", choose *re2*, and **Save** the changes.

In the pfSense® webGUI, go to **Interfaces > (new OPT interface)**, and **Enable** the interface. Give it an IP address in the same subnet as the modem, such as 192.168.1.5/24. Do not set a gateway. Rename the interface to *ModemAccess* or a similar useful name.

34.40.2 Configure NAT

Now NAT needs to be configured to translate traffic destined to the modem to the new interface. This is necessary so the modem sees the traffic sourced from an IP on its local subnet. Without this NAT, it would be necessary to configure a route on the modem so it knows how to reach the internal subnet. With some modems this isn't possible, and in most cases it's easier to NAT the traffic so routing isn't a concern. To add the NAT, browse to **Firewall > NAT**, and click the **Outbound** tab. Switch to **Manual Outbound NAT** and click **Save**. A rule for LAN to WAN is automatically added.

Click  to add a new Outbound NAT rule. For **Interface**, specify *ModemAccess*. For **Source**, specify *Network*, with the LAN subnet entered. The **Destination** is the IP subnet of the modem. In the **Translation** box, select *Interface Address*.

Then click **Save** and **Apply changes**.

It should now be possible to access the modem from LAN.

34.41 Configuring Multi-WAN for IPv6

Multi-WAN can be utilized with IPv6 provided that the firewall is connected to multiple ISPs or tunnels with static addresses.

See also:

See *Configuring IPv6 Through A Tunnel Broker Service* for help setting up a tunnel.

Gateway Groups work the same for IPv6 as they do for IPv4, but address families cannot be mixed within a group. A group must contain either *only* IPv4 gateways, or *only* IPv6 gateways.

Throughout this section "Second WAN" refers to the second or additional interface with IPv6 connectivity. It can be an actual interface that has native connectivity, or a tunnel interface when using a tunnel broker.

34.41.1 Caveats

In most cases, NAT is not used with IPv6 in any capacity as everything is routed. That is great for connectivity and for businesses or locations that can afford Provider Independent (PI) address space and a BGP peering, but it doesn't work in practice for small business and home users.

Network Prefix Translation (NPt) allows one subnet to be used for LAN which has full connectivity via its native WAN, but also has translated connectivity on the additional WANs so it appears to originate there. While not true connectivity for the LAN subnet via the alternate paths, it is better than no connectivity at all if the primary WAN is down.

Warning: This does not work for dynamic IPv6 types where the subnet is not static, such as DHCP6-PD.

34.41.2 Requirements

To setup Multi-WAN for IPv6 the firewall must have:

- IPv6 connectivity with static addresses on two or more WANs
- Gateways added to **System > Routing** for both IPv6 WANs, and confirmed connectivity on both.
- A routed /64 from each provider/path
- LAN using a static routed /64 or similar

34.41.3 Setup

The setup for IPv6 Multi-WAN is very close to the setup for IPv4. The main difference is that it uses NPt instead of NAT.

First, under **System > Routing** on the **Gateway Groups** tab, add Gateway Groups for the IPv6 gateways, with the tiers setup as desired. This works identically to IPv4.

Next, navigate to **System > General** and set one IPv6 DNS server set for each IPv6 WAN, also identically to IPv4.

Now add an NPt entry under **Firewall > NAT** on the **NPt** tab, using the following settings:

Interface Secondary WAN (or tunnel if using a broker)

Internal IPv6 Prefix The *LAN* IPv6 subnet

Destination IPv6 Prefix The second WAN routed IPv6 subnet

Note: This is **not** the /64 of the WAN interface itself – it is the /64 routed to the firewall on that WAN by the upstream.

What this does is akin to 1:1 NAT for IPv4, but for the entire subnet. As traffic leaves the second WAN, if it is coming from the LAN subnet, it will be translated to the equivalent IP address in the other subnet.

For example if the firewall has `2001:xxx:yyy::/64` on LAN, and `2001:aaa:bbb::/64` on the second WAN, then `2001:xxx:yyy::5` would appear as `2001:aaa:bbb::5` if the traffic goes out the second WAN. For more information on NPt, see *IPv6 Network Prefix Translation (NPt)*.

As with IPv4, the Gateway Groups must be used on LAN firewall rules. Edit the LAN rules for IPv6 traffic and set them use the gateway group, making sure to have rules for directly connected subnets/VPNs without a gateway set so they are not policy routed.

34.41.4 Alternate Tactics

Some users prefer to configure LAN with a "private" IPv6 subnet from the `fc00::/7` space and setup NPt for both WANs.

34.42 Configuring NAT for a VoIP PBX

For VoIP there are typically a few components to get right for proper inbound and outbound audio from a local PBX.

1. Port forward entries with firewall rules (Or 1:1 NAT with Firewall Rules)
2. Manual Outbound NAT with a rule at the top set to perform static port NAT on traffic from the PBX (Or 1:1 NAT)
3. On the PBX, ensure it is set properly for NAT with the correct external IP and local subnets defined.

34.42.1 Aliases to make it easy

It is easiest to start by making a few entries under **Firewall > Aliases** to make the rules easier to accomplish:

- Host alias for the PBX itself, named *PBX*, containing the local IP address of the PBX.
- Network or Host alias called *SIP_Trunks* for the upstream SIP trunk addresses, if known. If the *SIP_Trunk* address/network is not known or changes, do not make an alias and leave these values set to *any*.
- Port alias called *PBX_Ports* containing all of the port numbers needed for SIP, RTP, and other control ports. (usually *5060* and *10000:20000*, but varies from provider to provider and PBX implementation)

34.42.2 Port Forwards

For the port forward (**Firewall > NAT**, **Port Forwards** tab), it can be set as follows:

- **Interface**: *WAN*
- **Protocol**: *UDP* (or *TCP/UDP* if needed)
- **Source**: Type *Single Host or Alias*: *SIP_Trunks* – or a *Any* for the type if the SIP trunk IP addresses are not known.
- **Source Port**: *any/any*
- **Destination**: *WAN address* or external VIP for the PBX
- **Destination Port**: *PBX_Ports*
- **Redirect target IP**: *PBX*
- **Redirect target port**: *PBX_Ports*

34.42.3 Manual Outbound NAT

For *Manual Outbound NAT*, navigate to **Firewall > NAT**, **Outbound** tab, switch from *Automatic Outbound NAT* to *Manual Outbound NAT* and press **Save**. Then at the top of the list, create a rule that looks like so:

- **Interface**: *WAN*
- **Protocol**: *UDP*
- **Source**: *Network*, *PBX*
- **Source Port**: *[blank]*
- **Destination**: *Network*, *SIP_Trunks* – Or *Any* for the type if the SIP trunk IP addresses are not known
- **Destination Port**: *PBX_Ports* (or leave blank)

- **Translation**: *Interface address* if using the WAN IP address, or the external VIP for the PBX
- **Port**: *[blank]*
- **Static Port**: **CHECKED**

34.42.4 Reset States

After making the changes to NAT rules, the states for the PBX must be reset.

- Navigate to **Diagnostics > States**
- Enter the IP address of the PBX and click **Filter**
- Click **Kill**

Once the PBX re-registers it test inbound and outbound calls and confirm inbound and outbound audio works as expected.

34.43 Configuring NAT for VoIP Phones

If VoIP is being used, the default settings may not be correct in certain circumstances. The default settings handle the majority of scenarios, but depending on the specifics of a particular setup, changes may be necessary to obtain a working configuration.

The following sections will help to get local handsets working with a remote PBX. If the PBX is local and trying to communicate with a remote SIP trunk, see *Configuring NAT for a VoIP PBX* for more ideas.

34.43.1 Disable source port rewriting

By default pfSense® software rewrites the source port on all outbound traffic. This is necessary for proper NAT in some circumstances such as having multiple SIP phones behind a single public IP registering to a single external PBX. With a minority of providers, rewriting the source port of RTP can cause one way audio. In that case, setup manual outbound NAT and *Static Port* on all UDP traffic potentially with the exclusion of UDP 5060.

In old versions (pfSense 1.2.x and before) the firewall performed static port NAT on UDP 5060 traffic by default, but that is not desirable now because it breaks more scenarios than not currently. However, in cases where static port on UDP 5060 is required, configuring manual outbound NAT to perform static port NAT for udp/5060 will allow it to function.

34.43.2 Set Conservative state table optimization

The default UDP timeouts in pf are too low for some VoIP services. If phones mostly work, but randomly disconnect, set **Firewall Optimization Options** to *Conservative* under **System > Advanced**, **Firewall/NAT** tab.

A keep-alive or re-registration on the phone set for 20-30 seconds or so can also help, and is often a better solution.

34.43.3 Use the siproxd package

The *Siproxd package* is used only for deployments with local phones and a remote PBX where rewriting the source port breaks the ability to connect because the service will not work with rewritten source ports. In this very specific circumstance the siproxd package enables multiple phones to connect to a single outside server with a static source port of 5060.

Do not use this package if the PBX is local. Only use it if the upstream PBX strictly requires all phones to have a source port of 5060.

34.43.4 Disable scrub

In very rare circumstances, scrubbing needs to be disabled under **System > Advanced**, **Firewall/NAT tab**. In most cases this should be left at the default setting (unchecked). Only change this setting if it has been determined it is necessary to do so. Some phones send malformed packets that will be silently dropped without scrub active (e.g. unfragmented packets that claim to be fragmented).

34.44 Exporting NetFlow with softflowd

softflowd is a NetFlow collector that can be deployed on pfSense® software.

34.44.1 Installing softflowd

There is a package available under **System > Packages** on the **Available Packages** tab. Find it in the list, click  at the end of its row, and confirm the installation.

34.44.2 Configuring and Launching softflowd

Softflowd works similar to pfflowd.

Once the package has been installed, visit **Services > softflowd** to configure the service.

- **Interface**: Ctrl-click to select all of the interfaces from which NetFlow data should be gathered
- **Host**: The target NetFlow server which will receive flow data
- **Port**: The port on the **Host** which is listening for NetFlow data
- **Max Flows**: The number of flows to track before older flows expire
- **NetFlow Version**: The desired version of the NetFlow protocol. See NetFlow Versions on Wikipedia for more information.

34.44.3 Controlling softflowd from the Command Line

To view statistics about the running softflowd process, run the following command, replacing *em0* with the actual network interface to query:

```
: softflowctl -c /var/run/softflowd.em0.ctl statistics
```

To expire all flows and force an update to be sent to the netflow server, run the following command, replacing *em0* with the actual network interface to control:

```
: softflowctl -c /var/run/softflowd.em0.ctl expire-all
```

34.44.4 Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

34.44.5 Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

34.45 Bridging OpenVPN Connections to Local Networks

The OpenVPN configurations discussed to this point have all been routed, using *tun* interfaces. This is the preferable method, but OpenVPN also offers the option of using *tap* interfaces and bridging clients directly onto the LAN or other internal network. This can make the remote clients appear to be on the local LAN.

34.45.1 OpenVPN Server Settings

Most of the settings for a bridged remote access VPN are the same as above for a traditional remote access VPN. Only the differences will be noted here.

Device Mode To create a bridged connection, this must be set to *tap*.

Tunnel Network Remove values from the IPv4 Tunnel Network and IPv6 Tunnel Network boxes so they are empty. The way a *tap* bridge OpenVPN functions it does not need a tunnel network as OpenVPN doesn't use the same address assignment that it does for *tun* mode.

Bridge DHCP When selected, DHCP will be passed through to the bridged interface configured later. In the most common scenario, this is *LAN*. Using this method connecting clients would receive IP addresses from the same DHCP pool used by directly wired LAN clients.

Bridge Interface This setting does not create the bridge, it only indicates to OpenVPN which interface will be used for the bridge. In most cases, this is *LAN*. This controls which existing IP address and subnet mask are used by OpenVPN for the bridge. Setting this to *none* will cause the **Server Bridge DHCP** settings below to be ignored.

Server Bridge DHCP Start/End When using *tap* mode as a multi-point server, a DHCP range may be configured to use on the interface to which this *tap* instance is bridged. If these settings are left blank, DHCP will be passed through to the bridge interface, and the interface setting above will be ignored. This allows a range of IP addresses to be set aside for use only by OpenVPN clients so they may be contained within a portion of the internal network rather than consuming IP addresses from

the existing DHCP pool. Enter the **Server Bridge DHCP Start** and **Server Bridge DHCP End** IP address values as needed.

34.45.2 Creating the Bridge

Once the OpenVPN *tap* server has been created, the OpenVPN interface must be assigned and bridged to the internal interface.

Assign OpenVPN interface

In order to include the VPN interface in a bridge, it must be assigned. The procedure for assigning an interface is covered earlier in this chapter, in *Assigning OpenVPN Interfaces*.

Create Bridge

Once the VPN interface has been assigned, create the bridge as follows:

- Navigate to **Interfaces > Assignments**, **Bridges** tab
- Click  **Add** to create a bridge
- Ctrl-click both the VPN interface and the interface to which it will be bridged (e.g. *LAN*)
- Click **Save**

More information on bridging can be found in *Bridging*.

34.45.3 Connect with Clients

Clients connecting to the VPN must also be set to use *tap* mode. Once that has been set, connect with a client such as one exported using the OpenVPN Client Export package. The clients will receive an IP address inside the internal subnet as if they were on the LAN. They will also receive broadcast and multicast traffic.

34.46 Connecting to an OpenVPN Access Server

This guide will be a step by step walk through of how to get an OpenVPN client on pfSense® software connecting to OpenVPN AS (Access Server). The *.ovpn* file it generates is a bit odd so we will walk through how to extract the parts needed and where to put them into the pfSense software.

The guide will mainly focus on the pfSense side, but will touch on some basics of installing OpenVPN AS package.

34.46.1 OpenVPN AS vs Community OpenVPN

OpenVPN AS is the commercial version, that can be deployed via package on multiple Linux Distro's, Virtual Appliance or Cloud services like Amazon.

It comes with 2 licenses, so this makes handy for personal use be it simple package install on a low-end vps or deployed directly on a VPS AS appliance.

Why pay for a VPN connection, when a low-end VPS in pretty much any part of the world can be obtained for a year for quite often the cost of some VPN providers monthly fee.

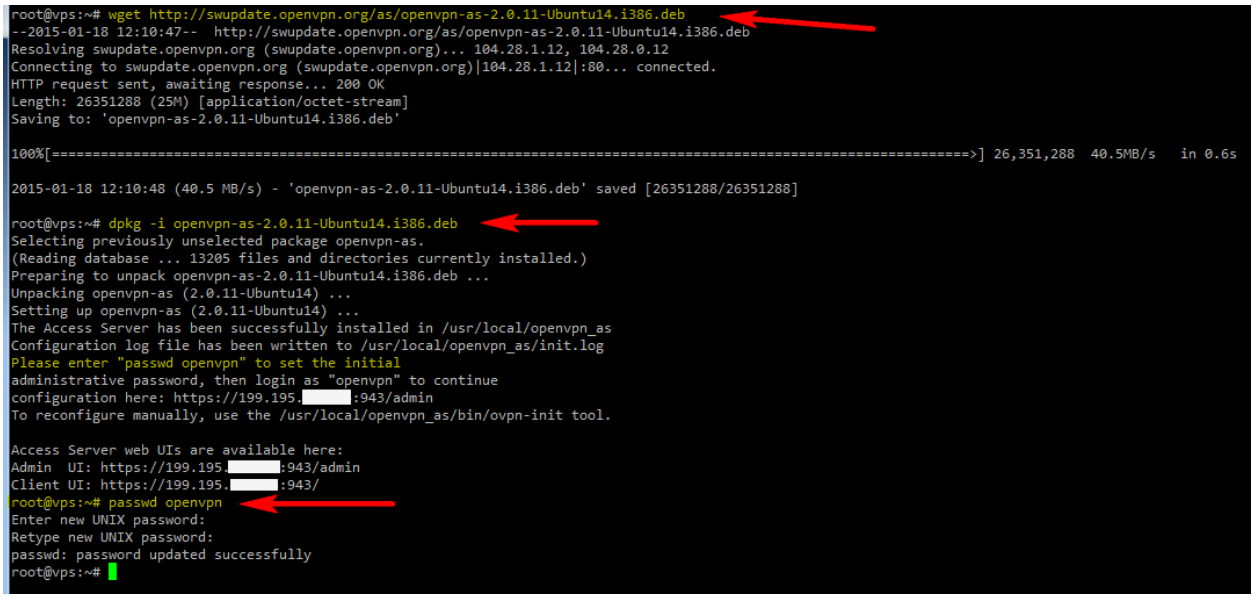
34.46.2 Getting and installing AS

Packages for favorite Linux Distros can be gotten here: https://openvpn.net/index.php/access-server/download-openvpn-as-sw.html

34.46.3 SSH to a VPS

In this example its a low end 128MB with Ubuntu 14.04 32bit. Make sure it has tun/tap VPN support - not all VPS types have this.

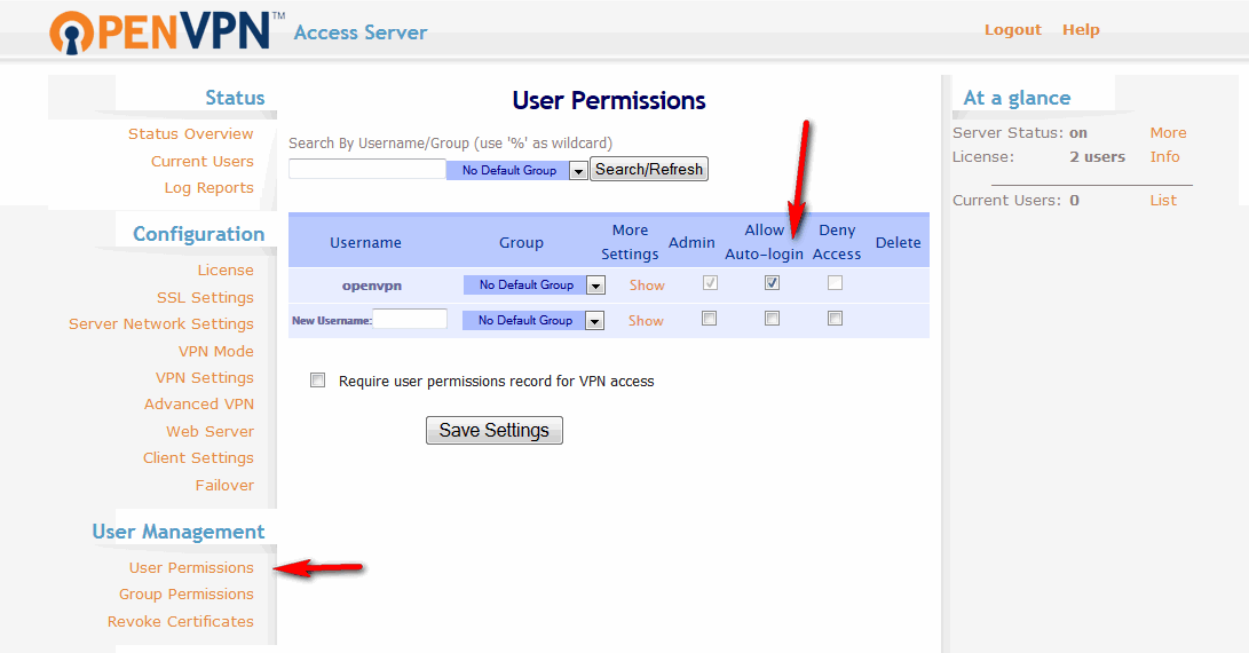
Logged in as root, use *wget* to grab the package, install the package with *dpkg* and then set the password on the OpenVPN account it creates.



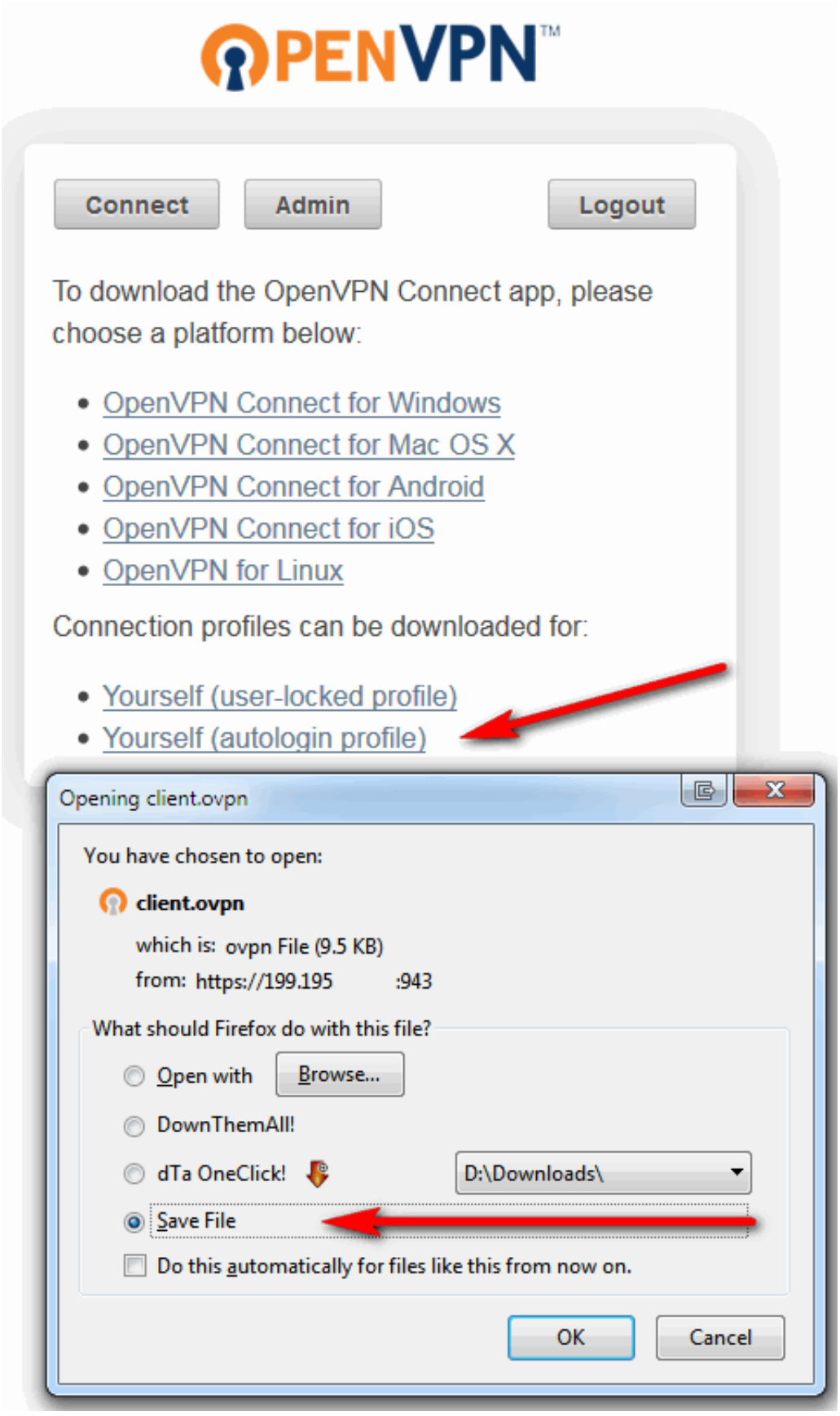
34.46.4 Go to the OpenVPN URL it lists

First go to the admin URL it provides - Set the profile to autologin and then grab the *.ovpn* file to extra info to put in to pfSense. Validate that the VPN works with either a normal OpenVPN client or the connect client that is available to download when the client navigates to the given URL and logs in as *admin*.

Login with the OpenVPN account with the changed password, then set permissions on the account to autologin. Make sure to save that and then it will ask to update the running server - click OK.



Logout and go back to the main non-admin URL and login again and the server will present the autologin profile to download


OPENVPN™
Connect
Admin
Logout
To download the OpenVPN Connect app, please choose a platform below:
OpenVPN Connect for Windows
OpenVPN Connect for Mac OS X
OpenVPN Connect for Android
OpenVPN Connect for iOS
OpenVPN for Linux
Connection profiles can be downloaded for:
Yourself (user-locked profile)
Yourself (autologin profile)
Opening client.ovpn
You have chosen to open:
client.ovpn
which is: ovpn File (9.5 KB)
from: https://199.195 :943
What should Firefox do with this file?
Open with
Browse...
DownThemAll!
dTa OneClick!
D:\Downloads\
Save File
Do this automatically for files like this from now on.
OK
Cancel

This is a good time to validate it is working by using that *.ovpn* configuration on a machine running the OpenVPN client or use the connect client downloaded at the same URL.

It is now possible to connect and go to a site that reports the client IP address, such as http://checkip.dyndns.org/, which should show the IP address of the VPN server to confirm that the client traffic is using the VPN. Once that is working, the pfSense router may be configured to use the info in the *.ovpn* file

34.46.5 Credits

Thanks to this great post https://forum.netgate.com/post/67545

34.47 Configuring a Single Multi-Purpose OpenVPN Instance

This recipe details one way to make a single OpenVPN server go a long way. Using this method access can be provided to a large pool of addresses for general access and then make use of some of the less intuitive features of OpenVPN to provide properly locked down access for various classes of user.

The end result is this:

- Single OpenVPN server instance listening on port 443/tcp
- A pool of addresses for general access on a single subnet
- A series of tiny address ranges (/30) that effectively allocate a static IP address to specific end users that can be easily grouped and firewalled

34.47.1 Setup

OpenVPN server

- Create the OpenVPN server as normal
- Set *TCP*, port *443*, and mode *tun*
- Set the **IPV4 Tunnel Network** as something similar to *10.33.249.0/24*
- Do not set **IPv4 Local Network(s)**.
- Set **Topology** to *net30*

The third octet should be a number far removed from VLAN/subnet numbers, a /24 is enough for most configurations.

This means that all connections will get an address from a global pool but they are useless unless access is allowed from that subnet in the firewall rules for the OpenVPN "interface"

Pick a subnet such as *10.33.250.0/24* which is not in use. This will be broken up into /30 mini subnets - one per client. If those run out, then start on *10.33.251.0/24*. Each of these new subnets needs a route in the main OpenVPN server Advanced settings, such as:

```
route 10.33.250.0 255.255.255.0;
```

OpenVPN certificate

Create a certificate in the usual way. I suggest setting the common name to first.last or company.first.last.

OpenVPN Client specific overrides

For each client create a **Client specific override**.

The tunnel networks will be **/30s** (i.e. One address for the network, one for the pfSense® OpenVPN server, one for the client and one for broadcast). So the first one will be *10.33.127.0/30* and the second one will be *10.33.127.4/30* and so on.

- Set the **Common Name** to first.last or what ever was used for the certificate
- **Description** - set to the Tunnel Network range, to make it easy to spot who has what
- **Tunnel Network** = last one allocated + 4 (see above)
- **Advanced** - *push "route 10.33.x.0 255.255.255.0";*

In the *route* above *x* is the customer network that this client may access.

Firewall rules

The client specific override forces a static IP onto the client which will be the third address in the range, for example:

```
10.33.250.8/30
10.33.250.9
10.33.250.10 - this is the static IP address for the client.
10.33.250.11
```

If there are several clients that access the same VLANs/subnets then put them together in an alias. Now add a rule on the OpenVPN tab of the Firewall rules granting access from the alias to the relevant subnets.

34.47.2 Notes

- Remember a client in this scheme needs to have a push route and a firewall rule to be be able access resources.
- It is recommended to allow ICMP everywhere on the OpenVPN firewall rules tab to help debugging.

Why use port 443/tcp ?

Listening on port 443/tcp is optional but can be useful. Many firewalls allow outbound access to destination port 443/tcp (https) or the ability of OpenVPN to go through web proxies may be utilized. There tend to be less problems using port 443. If there is only one external IP address available and need to run a web server on it then this will be impractical, use port 1194 in that case, or see *Sharing a Port with OpenVPN and a Web Server*. UDP is best for VPNs, but if port 443 is used, then use TCP.

34.48 Connecting OpenVPN Sites with Conflicting IP Subnets

One common use of NAT with OpenVPN is to mask conflicting LAN subnets between two locations. If two networks are using the exact same subnet, or overlapping subnets, as their LAN or other internal network they cannot communicate across a site-to-site VPN without NAT.

34.48.1 Site-to Site Example

In this example `10.3.0.0/24` is the LAN on both sides of a VPN. Hosts on the `10.3.0.0/24` subnet will never reach the other end of the VPN to communicate with the remote `10.3.0.0/24` subnet. Clients will always treat that network as local, attempting to reach the other systems via ARP. With NAT, however, the remote side can be made to function as if it were using a different subnet.

Note: Utilizing NAT will work for many protocols but some that are commonly desirable across VPN connections, primarily SMB/CIFS file sharing between Windows hosts, will not function in combination with NAT. If a protocol is used that is not capable of functioning with NAT, this is not a viable solution.

Figure *Site to Site with Conflicting Subnets* shows an example where both ends are using the same subnet. After assigning the OpenVPN interface to an OPT interface on both sides, as described in *Assigning OpenVPN Interfaces*, 1:1 NAT can be applied.

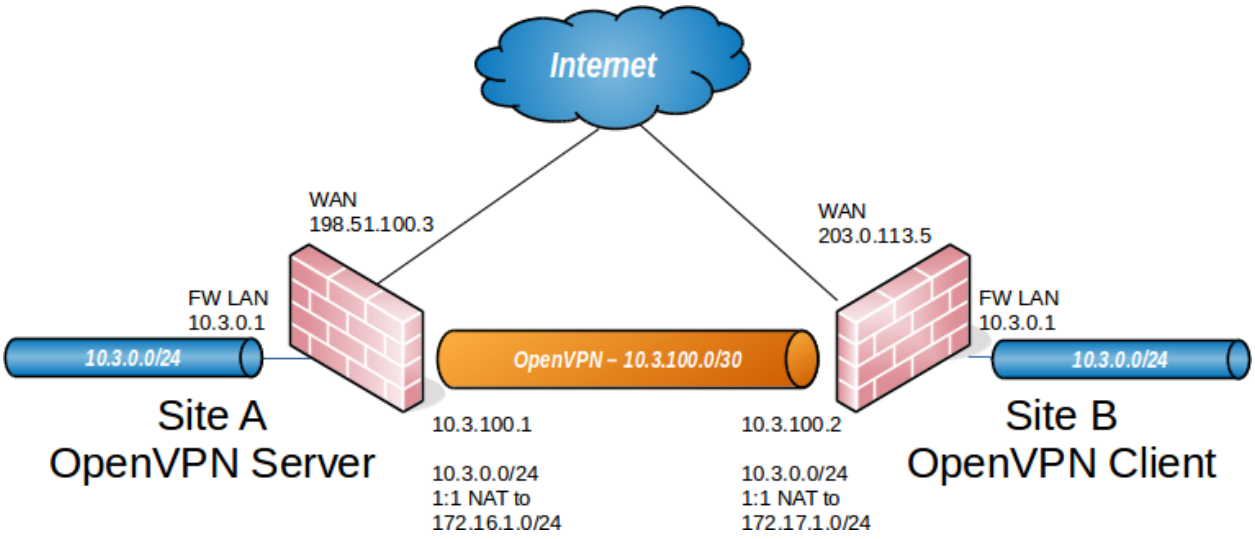


Fig. 67: Site to Site with Conflicting Subnets

The traffic from Site A will be translated to 172.16.1.0/24, and Site B will be translated to 172.17.1.0/24. A 1:1 NAT entry is added on each end to translate the entire /24 range. To reach Site A from Site B, 172.16.1.x IP addresses will be used. The last octet in the 10.3.0.x IP will be translated to the last octet in the 172.16.1.x translated IP. To reach 10.3.0.10 at Site A from Site B, use 172.16.1.10 instead. To reach 10.3.0.50 at Site B from Site A, use 172.17.1.50. Figure *Site B 1:1 NAT Configuration* show the 1:1 NAT configuration for each side, where the *tun* interface is assigned as OPT1.

In the OpenVPN configuration on both sides, the **Remote network** must be specified as the *translated* IP subnet, not as 10.3.0.0/24. In this example, the **Remote Network** at Site A is `172.17.1.0/24`, and `172.16.1.0/24` at Site B.

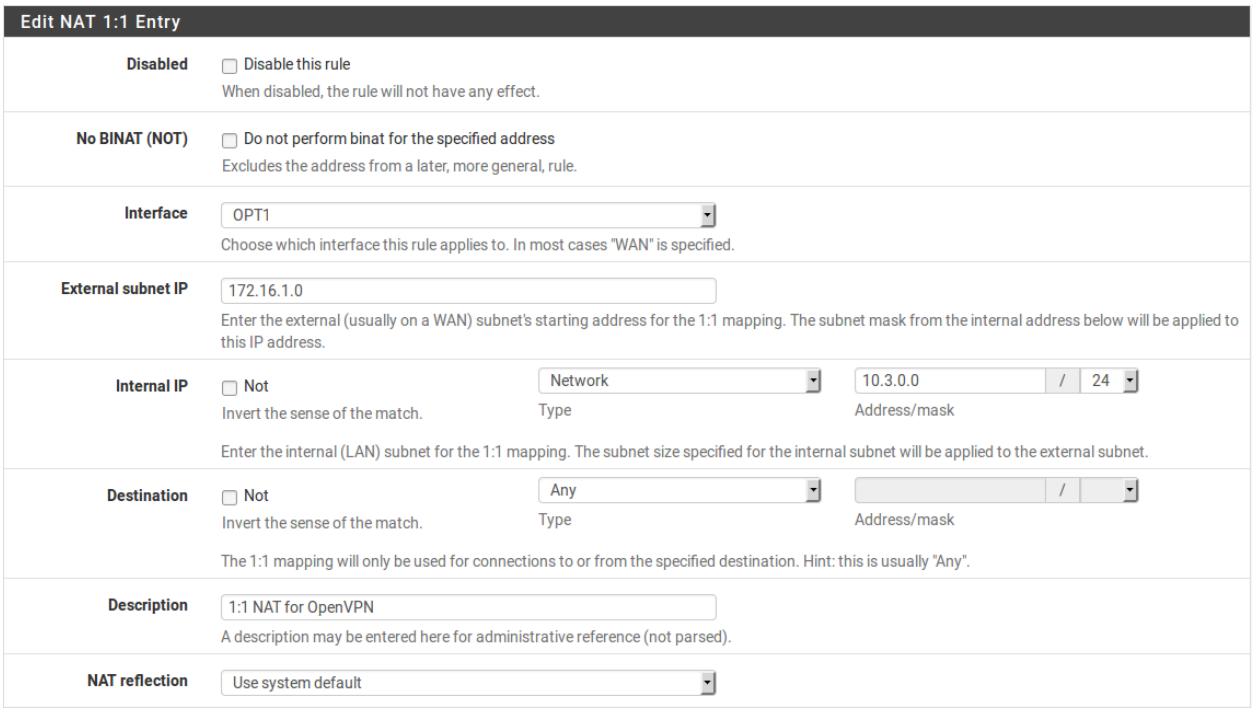


Fig. 68: Site A 1:1 NAT Configuration

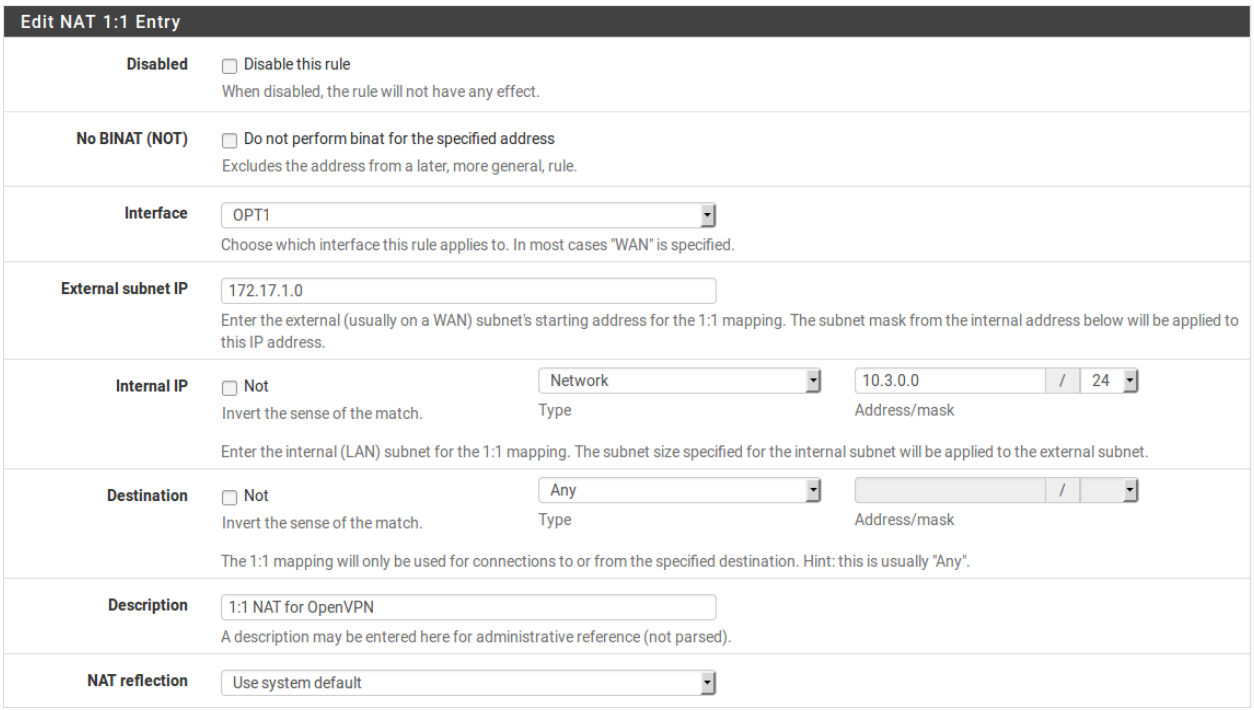


Fig. 69: Site B 1:1 NAT Configuration

After applying the NAT configuration changes and configuring the Remote network accordingly on both sides, the networks will be able to communicate using the translated subnets.

34.48.2 Site-to-Multi-Site Example

This section describes how to map multiple subnets that have the same IP address range using OpenVPN so that they can be accessed from a central site. For example 192.168.0/24 is a very common addressing scheme and the main site may need access to all the systems on those networks.

This is the desired outcome, Site 0 is "us":

```
Site 0 - 10.1.1/24
Site 1 - 192.168.0/24 -> 10.10.1/24
Site 2 - 192.168.0/24 -> 10.10.2/24
Site 3 - 192.168.0/24 -> 10.10.3/24
```

So we can now access 192.168.0.33 on say site 2 as 10.10.2.33.

This configuration requires 1:1 NAT and that means that some things may not work, see notes below.

There are multiple alternate ways to reach this goal. This example has the remote sites (Sites 1-3) run the "server" and Site 0 runs the "clients". Either direction works.

Preliminary

Pick **two** ranges for this scheme. The first one will be the mapping subnets and the second one will be for OpenVPN client to server connections. This example uses 10.10/16 for the mappings and 10.254.100/24 for connections. The first choice gives 253 mappings and the second choice gives 64 /30 subnets to link it all up.

Examples before NAT:

Site 0 to Site 1:

```
            <--------              (OpenVPN)             ---------->

10.1.1/24 --- 10.254.100.2/30 --- INTERNET --- 10.254.100.1/30 --- 192.168.0/24
```

Site 0 to Site 2:

```
            <--------              (OpenVPN)             ---------->

10.1.1/24 --- 10.254.100.6/30 --- INTERNET --- 10.254.100.5/30 --- 192.168.0/24
```

Note: Notice how the addresses on the right are still non unique - they are still always 192.168.0/24.

We use NAT to make each remote site unique:

```
10.1.1/24 --- (OpenVPN) --- INTERNET --- OpenVPN_Interface - NAT - 10.10.1/24 <=> 192.
↪168.0/24
10.1.1/24 --- (OpenVPN) --- INTERNET --- OpenVPN_Interface - NAT - 10.10.2/24 <=> 192.
↪168.0/24
10.1.1/24 --- (OpenVPN) --- INTERNET --- OpenVPN_Interface - NAT - 10.10.3/24 <=> 192.
↪168.0/24
```

Site 0 throws packets to a non existent destination (the mapping subnet) down the tunnel that corresponds to the desired Site 1,2 or 3 and the NAT at the other end translates to and from that mapping and sends the result back to Site 0.

Recipe

In the following examples, only necessary changes from default are given. Pick suitable transports (UDP by default) and ports (1194 by default) Put in a suitable firewall rule on the server WAN interfaces to allow the inbound connection VPN. This should be restricted to the WAN IP address of Site 0. Both ends must have a suitable firewall rule on the OpenVPN interface for traffic to pass.

At Site 1-3

OpenVPN server

At each remote site, create a new OpenVPN server:

```
Server Mode:          Peer to Peer
Description:          Link to Site 0
TLS authentication:   Tick "Automatically generate a shared TLS authentication key."
IPv4 Tunnel Network:  <link subnet> eg 10.254.100.0/30
IPv4 Remote networks: <IP range(s) at Site 0> eg 10.1.1.0/24
```

If Site 0 has multiple IP ranges then specify them all in IPv4 Remote networks, comma separated. After saving, copy the key ("2048 bit OpenVPN static key") that was generated to the other end (see below)

1:1 NAT

Add one of these for *each* network range at Site 0:

```
Interface:          OpenVPN
External subnet IP: <mapping subnet, first IP> eg 10.10.1.0
Internal IP:        LAN net
Destination:        Network, <IP range on Site 0> eg 10.1.1.0/24
```

Not the default gateway

If this firewall is not the default gateway for the site then use an outbound NAT rule on LAN to ensure that replies from the clients return via the OpenVPN tunnel. Without this, the systems at Sites 1-3 will reply via their default gateway because they will be unaware of the Site 0 network. Another option is to put a suitable route on the site gateway via the LAN address of the OpenVPN system but this will introduce an asymmetric route and which will potentially break things even more than the double NAT.

At Site 0

OpenVPN client

Create a separate OpenVPN client for each remote subnet (Where examples are given they are for Site 1):

```
Server mode:            Peer to Peer (Shared key)
Server host or address: <ip.add.re.ss of the other end>
Description:            <NAME mapping_subnet link_subnet> eg Site 1 10.254.100.0/30␣
↪10.10.1.0/24
Shared Key:             <Copy from the server for the site link>
```

(continues on next page)

(continued from previous page)

```
IPv4 Tunnel Network:    <link subnet> eg 10.254.100.0/30
IPv4 Remote networks:   <mapping network> eg 10.10.1.0/24
```

Notes

- SIP and RTP for example will be tricky
- DNS may not work well under this scheme
- Anything relying on DNS eg web links that don't use the passed host name but use the built in name
- Don't forget to put suitable firewall access rules on the various OpenVPN interfaces

34.49 OpenVPN Remote Access Configuration Example

The OpenVPN wizard is a convenient way to setup a remote access VPN for mobile clients. It configures all of the necessary prerequisites for an OpenVPN Remote Access Server:

- An authentication source (Local, RADIUS server, or LDAP server)
- A Certificate Authority
- A Server Certificate
- An OpenVPN server instance.

By the end of the wizard a fully functioning sever will be configured and ready for users. An example setup will be used to aide in explaining the options available in the wizard.

34.49.1 Before Starting The Wizard

Before starting the wizard to configure the Remote Access Server, there are some details that must be planned.

Determine an IP addressing scheme

An IP subnet must be chosen for use by the OpenVPN clients themselves. This is the subnet filled in under **Tunnel Network** in the server configuration. Connected clients will receive an IP address within this subnet, and the server end of the connection also receives an IP address used by the client as its gateway for networks on the server side.

As always when choosing internal subnets for a single location, ideally the chosen subnet will be designed so that it can be CIDR summarized with other internal subnets. The example network depicted here uses 10.3.0.0/24 for LAN, and 10.3.201.0/24 for OpenVPN. These two networks can be summarized with 10.3.0.0/16, making routing easier to manage. CIDR summarization is discussed further in *CIDR Summarization*.

Example Network

Figure *OpenVPN Example Remote Access Network* shows the network configured in this example.

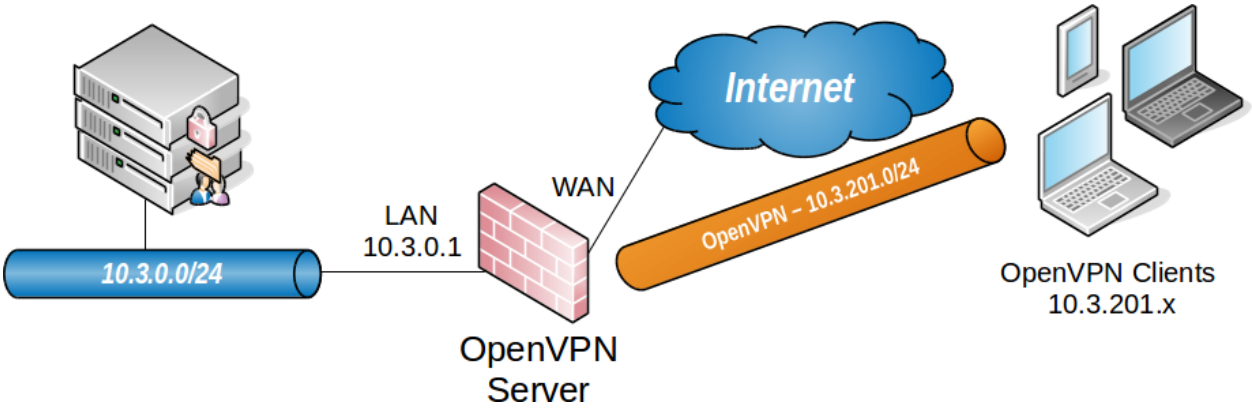


Fig. 70: OpenVPN Example Remote Access Network

34.49.2 Choose Authentication Type

On the first screen of the OpenVPN Remote Access server wizard, choose a method for user authentication. The choices available for **Authentication Backend Type** are *Local User Access*, *LDAP*, and *RADIUS*.

If an existing authentication system is already in place, such as Active Directory, pick *LDAP* or *RADIUS* depending on how that system is configured. *Local User Access* may be selected to manage the users, passwords, and certificates on the pfSense® firewall. When using *Local User Access*, per- user certificates may be used easily, managed completely in the pfSense GUI. This is much more secure, but depending on the number of users which will access the service, may be less convenient than using a central authentication system.

Note: For *LDAP* or *RADIUS*, per-user certificates cannot be used without generating them manually.

The *Local User Access* choice is the equivalent of choosing *Remote Access (SSL/TLS + User Auth)* mentioned earlier in this chapter. *LDAP* and *RADIUS* are equivalent to *Remote Access (User Auth)*.

After selecting the authentication server type, click *Next*. If *LDAP* or *RADIUS* were chosen the server configuration for those choices will be the next step. If *Local User Access* was chosen, the LDAP and RADIUS wizard steps are skipped. For this example, *Local User Access* will be chosen, but the other options are discussed for completeness.

34.49.3 Choosing an LDAP Server

If an LDAP server is already defined on the pfSense firewall it may be chosen from the list. To use a different LDAP server instead choose **Add new LDAP server**. If there are no LDAP servers defined, this step is skipped.

34.49.4 Adding an LDAP Server

If no LDAP servers exist or **Add new LDAP server** is chosen a screen will be presented with the options needed to add a new server. Many of these options will depend on the specific LDAP directory configuration and structure. If there is any uncertainty about the settings, consult the LDAP server administrator, software vendor, or documentation.

Note: The details of LDAP servers are covered in *Authentication Servers*. Some detail is omitted here since the options are discussed in-depth elsewhere. For more information on the options listed in this section, refer there instead.

Name Descriptive name for this LDAP server, for reference.

Hostname or IP address The hostname or IP address of the LDAP server.

Port The port on which the LDAP server may be contacted. The default port is `389` for standard TCP connections, and `636` for SSL.

Transport This can be set to *TCP - Standard* for unencrypted connections, or *SSL - Encrypted* for secure connections. A standard connection may be sufficient at least for local servers or initial testing. If the server is remote or crosses any untrusted network links, SSL is a more secure choice. If SSL is to be used, the CA Certificate from the LDAP server must be imported into pfSense, and the **Hostname or IP address** above must match the value in the **Common Name** field of the server certificate.

Search Scope Level Selects how deep to search in the LDAP directory, *One Level* or *Entire Subtree*. Most commonly, *Entire Subtree* is the correct choice.

Search Scope Base DN The Distinguished Name upon which the search will be based. For example `DC=example,DC=com`

Authentication Containers These values specify where in the directory that users are found. For example, it may be `CN=Users;DC=example`.

LDAP Bind User DN The Distinguished Name for a user that can be used to bind to the LDAP server and perform authentication. If this is left blank, an anonymous bind will be performed, and the password setting below will be ignored.

LDAP Bind Password The password to be used with the LDAP Bind User DN.

User Naming Attribute Varies depending on the LDAP directory software and structure. Typically `cn` for OpenLDAP and Novell eDirectory, and `samAccountName` for Microsoft Active Directory.

Group Naming Attribute Varies depending on the LDAP directory software and structure, but is most typically `cn`.

Member Naming Attribute Varies depending on the LDAP directory software and structure. Typically `member` on OpenLDAP, `memberOf` on Microsoft Active Directory, and `uniqueMember` on Novell eDirectory.

34.49.5 Choosing a RADIUS Server

If there is an existing RADIUS server defined on the pfSense firewall, choose it from the list. To use a different RADIUS server, instead choose **Add new RADIUS server**. If no RADIUS servers are defined on pfSense, this step is skipped.

34.49.6 Adding a RADIUS Server

If no RADIUS servers exist, or **Add new RADIUS server** was selected, a screen is presented with the options needed to add a new server. If there is any uncertainty about the settings, consult the RADIUS server administrator, software vendor, or documentation.

Note: The details of RADIUS servers are covered in *Authentication Servers*. Some detail is omitted here since the options are discussed in-depth elsewhere. For more information on the options listed in this section, refer there instead.

Name Descriptive name for this RADIUS server, for reference.

Hostname or IP address The hostname or IP address of the RADIUS server.

Authentication Port Port used by the RADIUS server for accepting Authentication requests, typically `1812`.

Shared Secret The **Shared Secret** is the password configured on the RADIUS server for accepting authentication requests from the IP address of the pfSense firewall.

34.49.7 Choosing a Certificate Authority

If there is an existing Certificate Authority defined on the pfSense firewall, it may be chosen from the list. To create a new Certificate Authority, choose **Add new CA**. If no Certificate Authorities are defined, this step is skipped.

34.49.8 Creating a Certificate Authority

This step presents all of the necessary fields to create a new certificate authority (CA). Every option on this page is required, and all fields must be filled out correctly to proceed. The CA is used to establish a trust base from which the server certificates can be generated and deemed "trustworthy" by clients. Because this CA is self-generated, it will only be trusted by clients who are also supplied with a copy of this CA certificate.

See also:

For more information on creating and managing CAs, see *Certificate Authority Management*.

Descriptive Name A name for reference to identify this certificate. This is the same as Common Name field for other Certificates. For this example CA, `ExampleCoCA` is used. Although using spaces in this field is allowed, we strongly discourage using spaces in a Common Name field because some clients have issues handling them properly.

Key Length Size of the key which will be generated. The larger the key, the more security it offers but larger keys are generally slower to use. `2048` is a good choice.

Lifetime The time in days that this CA will be valid. On a self-generated CA such as this, it is commonly set to `3650`, which is approximately 10 years.

Country Code Two-letter ISO country code (e.g. US, AU, CA). If the two-letter ISO country code is unknown, locate it on the ISO Online Browsing Platform site. Since the ExampleCo company is set in the United States, enter `US` for this example.

State or Province Full unabbreviated State or Province name (e.g. Texas, Indiana, California). ExampleCo is located in `Texas` for this example.

City City or other Locality name (e.g. Austin, Indianapolis, Toronto). ExampleCo's headquarters is in `Austin`.

Organization Organization name, often the Company or Group name. `ExampleCo` goes here for this example. **Do not use any special characters in this field, not even punctuation such as a period or comma**.

E-Mail E-mail address for the Certificate contact. Often the e-mail of the person generating the certificate, such as `vpnadmin@example.com`.

Click **Add new CA** to finish the CA creation process

34.49.9 Choosing a Server Certificate

If there is an existing Certificate defined on the pfSense firewall, it may be chosen from the list. To create a new Certificate, choose **Add new Certificate**. If no Certificates are defined, this step is skipped.

34.49.10 Adding a Server Certificate

This screen creates a new server certificate which will be used to verify the identity of the server to the clients. The server certificate will be signed by the certificate authority chosen or created previously in the wizard. In most cases, as with this example, the same information from the previous step is used and it will be pre-filled on the form automatically.

Descriptive Name This is the Common Name (CN) field for the server certificate and is also used to reference the certificate in pfSense. Using the hostname of the firewall is a common choice for a server certificate, such as `vpn.example.com`. Although using spaces in this field is allowed, we strongly discourage using spaces in a Common Name field because clients tend to have issues handling them properly.

Key Length Size of the key which will be generated. The larger the key, the more security it offers but larger keys are generally slower to use. `2048` is a good choice.

Lifetime Lifetime in days. This is commonly set to `3650` (Approximately 10 years).

Country Code Two-letter ISO country code (e.g. US, AU, CA)

State or Province Full State of Province name, not abbreviated (e.g. Texas, Indiana, Ontario).

City City or other Locality name (e.g. Austin, Indianapolis, Toronto).

Organization Organization name, often the Company or Group name. **Do not use any special characters in this field, not even punctuation such as a period or comma**.

E-Mail E-mail address for the Certificate contact. Often the e-mail of the person generating the certificate. (e.g. `vpnadmin@example.com`)

Click **Create New Certificate** to store the settings and continue to the next step of the wizard.

34.49.11 Configuring OpenVPN Server Settings

The options on this step of the wizard configure each aspect of how the OpenVPN server itself will behave as well as options which are passed on to clients. The options presented here are the same as those discussed previously in *OpenVPN Configuration Options*, refer to that section for details. Because the options are covered in detail in that section, only the settings for this example will be mentioned.

General OpenVPN Server Information

These options control how the OpenVPN instance operates.

Interface Since incoming connections will be from the WAN side, select *WAN*.

Protocol The default of *UDP* is acceptable.

Local Port This will be the first OpenVPN server instance so the default of `1194` is preferred. If there is an existing OpenVPN on that port, use a different port number. The wizard will suggest an unused port number.

Description As this will be for remote user access, `ExampleCo Mobile VPN Clients` is a fitting description.

Cryptographic Settings

These options control how traffic in the tunnel is encrypted and authenticated.

TLS Authentication TLS is highly desirable so check **Enable authentication of TLS packets**.

Generate TLS Key There is no existing TLS key, so check **Automatically generate a shared TLS authentication key**.

TLS Shared Key Since there is no existing TLS key, leave this blank.

DH Parameters Length Select *2048*, as it is good balance of speed and strength.

Encryption Algorithm This can be left at the default value of *AES-128-CBC*, but any other option would also work well as long as the clients are set to match.

Auth Digest Algorithm Leave at the default *SHA1 (160-bit)*

Hardware Crypto The target device has no accelerator, so leave this set to *No Hardware Crypto Acceleration*

Tunnel Settings

These options control how traffic coming from the remote clients will be routed.

Tunnel Network As in the diagram at the start of this example, the subnet `10.3.201.0/24` has been chosen for the VPN clients.

Redirect Gateway For ExampleCo's setup, The VPN will only carry traffic which is destined for the subnets at the main office so this box is left *unchecked*.

Local Network This is the main office subnet, which in this example is `10.3.0.0/24`.

Concurrent Connections ExampleCo does not want to limit the number of clients which can connect at the same time, so this is left blank.

Compression To improve throughput of traffic on the VPN tunnel at the expense of some CPU power, this is set to *Enabled with Adaptive Compression*.

Type-of-Service This box is unchecked, as there is no traffic on this VPN which requires prioritization/QoS.

Inter-Client Communication Because the clients on this VPN have no need to connect to other client machines, this box is unchecked.

Duplicate Connections Because unique certificates exist for every client, this is unchecked.

Client Settings

These options control specific settings given to the clients when a connection is established.

Dynamic IP The clients will connect from all over the country and unknown mobile networks and their IP addresses are likely to change without notice so this option is checked.

Address Pool The clients will be assigned addresses from the tunnel network above, so this is checked.

Topology The method used to assign IP addresses to clients. The default of *Subnet* is the best choice.

DNS Default Domain Enter the domain for ExampleCo here, `example.com`.

DNS Servers Any internal DNS server could be used here. ExampleCo has a Windows Active Directory Domain Controller which is configured to act as a DNS server, `10.3.0.5`.

NTP Servers The server above, `10.3.0.5`, is also used to synchronize client PC clocks.

NetBIOS Options Clients will need access to Windows shares behind the VPN, so check **Enable NetBIOS over TCP/IP**.

NetBIOS Node Type Because DNS is used primarily, select *h-node*.

NetBIOS Scope ID This will be left blank, since the NetBIOS scope is not limited.

WINS Servers WINS has been deprecated, so this is left blank.

Advanced At this time no additional tweaks are needed, so this is left blank.

34.49.12 Firewall Rule Configuration

As with other parts of the firewall, by default all traffic is blocked from connecting to VPNs or passing over VPN tunnels. This step of the wizard adds firewall rules automatically to allow traffic to connect to the VPN and also so connected clients can pass traffic over the VPN.

Traffic from clients to server

Check this box to add a firewall rule on the chosen interface for the tunnel (e.g. *WAN*) which lets clients connect. It allows all clients from any source address to connect by default. To allow connections from a limited set of IP addresses or subnets, either make a custom rule or check this box and alter the rule it creates. Since in this example clients are connecting from all over the country, the rule created by this checkbox is ideal, so the box is checked.

Traffic from clients through VPN tunnel

This setting allows all traffic to cross the OpenVPN tunnel, which is desirable for this example, so this box is checked.

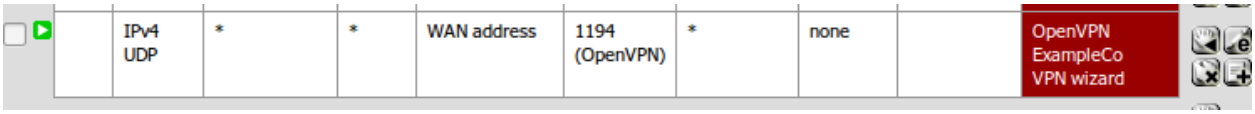
34.49.13 Finishing the Wizard

Click **Finish** and the wizard is now complete; The tunnel is fully configured and ready for client connections. From here the next steps are to add users and configure client devices. If adjustments to the automatically generated firewall rules are required, make them now.

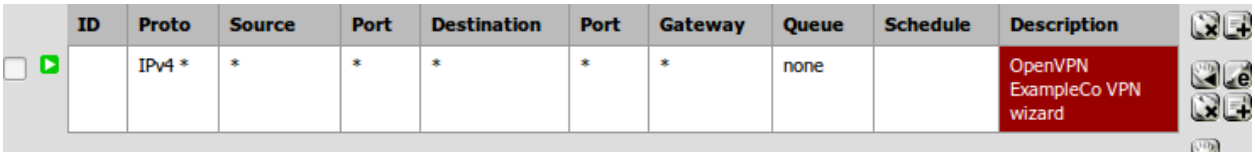
34.49.14 Verifying the Setup

Look at firewall rules (**WAN** and **OpenVPN** tabs)

- **WAN** tab rule should pass from any to the *OpenVPN* port on the *WAN address*



- **OpenVPN** tab rule should allow anything from any/to any



34.49.15 Adjustments

Some settings are not presented in the wizard but might be a better fit for some situations than the defaults chosen by the wizard.

Server Mode

The OpenVPN **Server Mode** allows selecting a choice between requiring Certificates, User Authentication, or both. The wizard defaults to *Remote Access (SSL/TLS + User Auth)*. The possible values for this choice and their advantages are:

- *Remote Access (SSL/TLS + User Auth)*
  - Requires both certificates AND username/password
  - Each user has a unique client configuration that includes their personal certificate and key.
  - Most secure as there are multiple factors of authentication (TLS Key and Certificate that the user has, and the username/password they know)
- *Remote Access (SSL/TLS)*
  - Certificates only, no auth
  - Each user has a unique client configuration that includes their personal certificate and key.
  - Useful if clients should not be prompted to enter a username and password
  - Less secure as it relies only on something the user has (TLS key and certificate)

- *Remote Access (User Auth)*
  - Authentication only, no certificates
  - Useful if the clients should not have individual certificates
  - Commonly used for external authentication (RADIUS, LDAP)
  - All clients can use the same exported client configuration and/or software package
  - Less secure as it relies on a shared TLS key plus only something the user knows (Username/password)

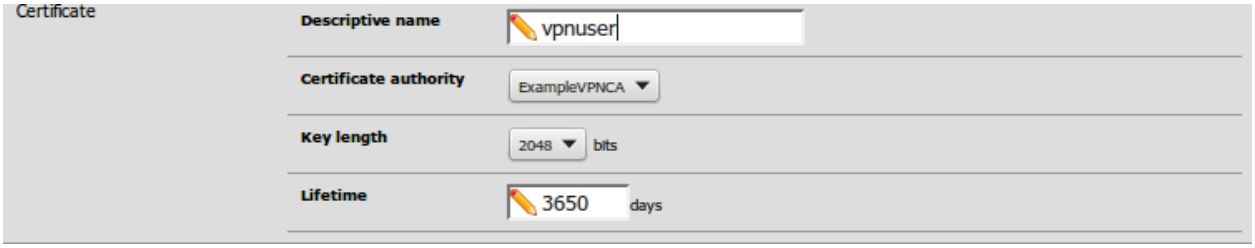
Certificate Revocation

Compromised certificates can be revoked by creating a Certificate Revocation List (CRL) in **System > Cert Manager** on the **Certificate Revocation** tab, adding the certificate to it, and then selecting that CRL on the OpenVPN server settings.

34.49.16 Adding a User with a Certificate

If the mode has been left at the wizard's default or on a mode that includes local user authentication, a user must be created in the user manager.

- Navigate to **System > User Manager**
- Click  To add a user
- Fill in **Username**
- Fill in **Password / Confirm password**
- Check **Click to create a user certificate**.
- Fill in the **Descriptive Name** as the *username*



- Choose the appropriate **Certificate Authority**
- Click **Save**

34.49.17 OpenVPN Client Export Package

The OpenVPN Client Export Package allows exporting configurations formatted for a wide variety of platforms. It also allows exporting a pre-packaged Windows installer executable which includes the configuration bundled inside for a painless client installation.

See *OpenVPN Client Export Package* for more.

34.50 Authenticating OpenVPN Users with FreeRADIUS

Using OpenVPN with the *FreeRADIUS package*.

34.50.1 Purpose

This document demonstrates how to setup OpenVPN while allowing for authentication via RADIUS. Usernames and Passwords can be managed centrally on the firewall, and additional RADIUS-specific options may be used. This is a plus because login times, access limits, and other options are possible.

34.50.2 Requirements

- A working OpenVPN server. See *OpenVPN Remote Access Configuration Example*
- FreeRADIUS Installed. See *FreeRADIUS package*

34.50.3 Add an interface to FreeRADIUS

- Navigate to **Services > FreeRADIUS**
- Select the **Interfaces** tab
- Click + to add a new entry
- Enter `*` for the **Interface IP Address**, or `127.0.0.1` to bind only to Localhost
- Enter `1812` for the **Port**
- Select *Authentication* for the **Interface Type**
- Click **Save**

34.50.4 Add a NAS client to FreeRADIUS

- Navigate to **Services > FreeRADIUS**
- Select the **NAS / Clients** tab
- Click + to add a new entry
- Enter `127.0.0.1` in the **Client IP Address** field
- Enter `pfSense`, `OpenVPN`, or similar in the **Client Shortname** field
- Enter a random/long password in the **Client Shared Secret** field
- Enter a **Description** that will help identify this connection.
- Click **Save**

34.50.5 Add Users

- Navigate to **Services > FreeRADIUS**
- Select the **Users** tab.

  This is where *every* user to authenticate with FreeRadius/OpenVPN is managed
- Click + to add a new entry
- Enter a **Username** and **Password**
- Enter any additional desired options, such as **Number of simultaneous connections**
- [optional] Set the **Session Timeout** When this timer expires, the user will be kicked off and will have to login again
- Repeat as needed for additional users
- Click **Save**

34.50.6 Configure a pfSense Authentication Server

- In the pfSense® webGUI, navigate to **System > User Manager**
- Select the **Servers** tab
- Click + to add a new entry
- Enter a **Descriptive name** such as `FreeRADIUS`
- Select *RADIUS* for the **Type**
- Enter `127.0.0.1` for the **Hostname or IP address**
- Enter the password created above for **Shared Secret**
- Select *Authentication* for **Services offered**
- Enter `1812` for **Authentication port value**
- Click **Save**

34.50.7 Test RADIUS Authentication

- Navigate to **Diagnostics > Authentication**
- Select the authentication server entered above
- Fill in a **Username** and **Password** configured in FreeRADIUS
- Click **Test**

If the test succeeded, continue. Otherwise, see the Troubleshooting section below.

34.50.8 Configure OpenVPN to use RADIUS

- Navigate to **VPN > OpenVPN**
- Select the **Servers** tab
- Edit the existing Remote Access server
- Ensure that the **Mode** is either **Remote Access (User Auth)** or **Remote Access (SSL/TLS + User Auth)**
- Select *FreeRADIUS* or the **Descriptive Name** chosen above for the FreeRADIUS authentication server in the **Backend for authentication** field.
- Click **Save**

34.50.9 Troubleshooting

Sometimes things don't work as expected. The following options can be helpful in troubleshooting FreeRADIUS and OpenVPN. Commands must be run at a shell prompt either via the console or via SSH unless otherwise specified.

- Make OpenVPN more Verbose and force it to log to a non-standard location so it can be read it easier.
  - Navigate to **VPN > OpenVPN** and select the server
  - Change **Verbosity level** to *7*

    This will log everything from OpenVPN to the **OpenVPN** tab under **Status > System Logs**. It can be watched with the following command (while trying to connect/etc):

    ```
    clog -f /var/log/openvpn.log
    ```

- FreeRADIUS may also be watched for attempted connections/authorizations (Failed or successful):

  ```
  clog -f /var/log/system.log
  ```

- With this information in hand, Google and the Netgate Forum can be a very good resource.

Adapted from / Previously reprinted with permission from http://www.fusionnetwork.us/index.php/component/content/article/15-general-tutorials/23-pfsense-openvpn-freeradius

34.51 Authenticating OpenVPN Users with RADIUS via Active Directory

This recipe demonstrates setting up OpenVPN on pfSense® software for Windows clients, using certificates with user authentication via RADIUS in Active Directory.

The target audience of this recipe is small businesses that want to roll out secure VPN connectivity for their users using free software. Due to the nature of its set up, which is mostly manual, this process may be too inefficient for larger businesses.

34.51.1 Versions

- pfSense software version 2.x
- Active Directory on Windows Server 2008 R2 - I'm using a Forest Functional Level of 2008 R2 but I don't think that's really a prerequisite. If it doesn't work, user account passwords may need to be stored using reversible encryption but since that is a serious security issue, it is better to upgrade to at least 2008 R2.

34.51.2 On security and a disclaimer

I am not a security expert. However the method described in this article is they way it should be:

- Two-factor authentication: something the user has (the installed certificate) and something the user knows (AD user account name and password);
- The connection is encrypted and nothing crosses the Internet in plain text.

If a laptop gets stolen, no one can dial into the corporate network if they don't know a username and password. If someone guesses a password, they will also need the certificate to dial in.

I can not guarantee that no bad things happen because of following this recipe. Please consult other sources, use common sense and try breaking into the system to check if it's safe.

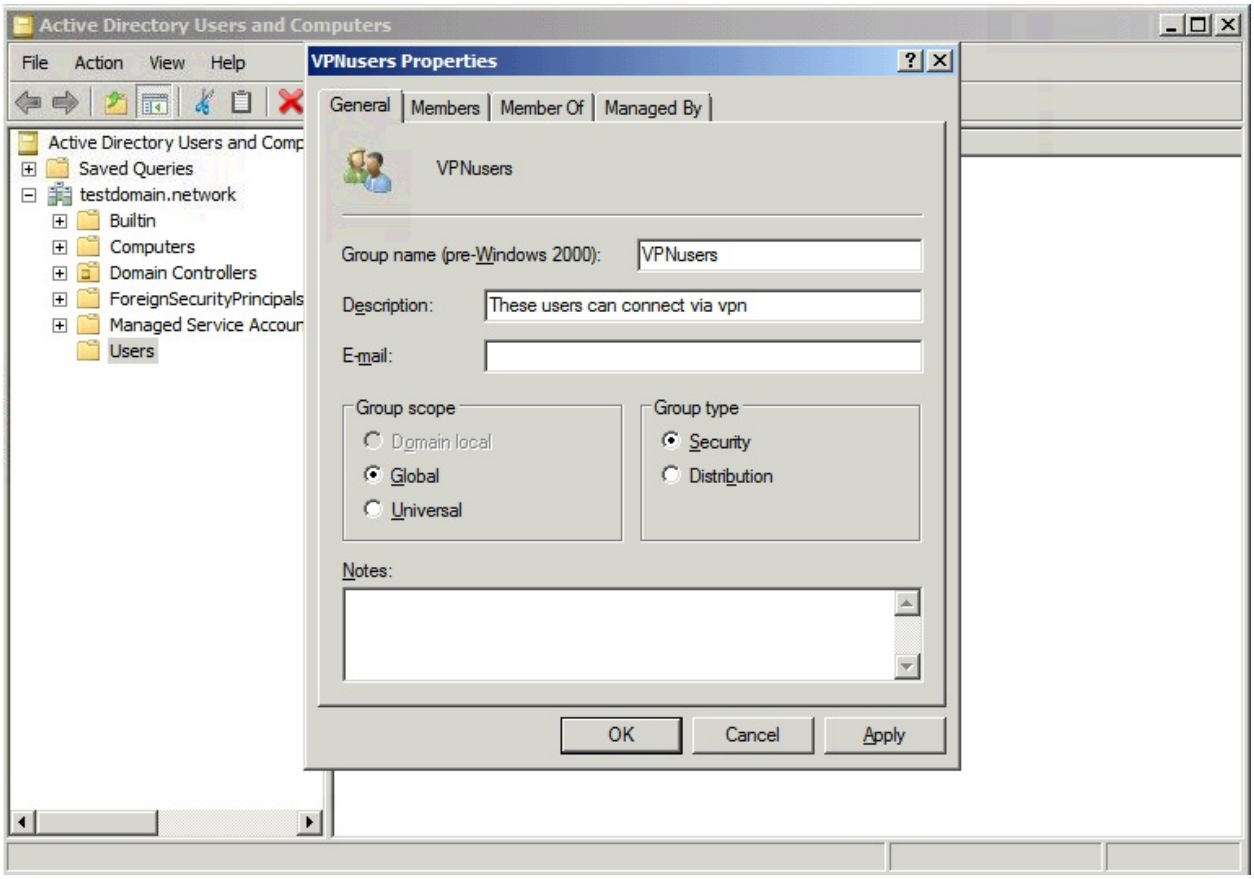
34.51.3 Thanks

Thanks to the pfSense forum, in particular to user unguzov, who wrote a shorter version of this recipe. I adapted his version and added screenshots. Thanks to Evan Jensen for providing some English version screenshots. Thanks to Dan, who alerted me on the question of the policy order.

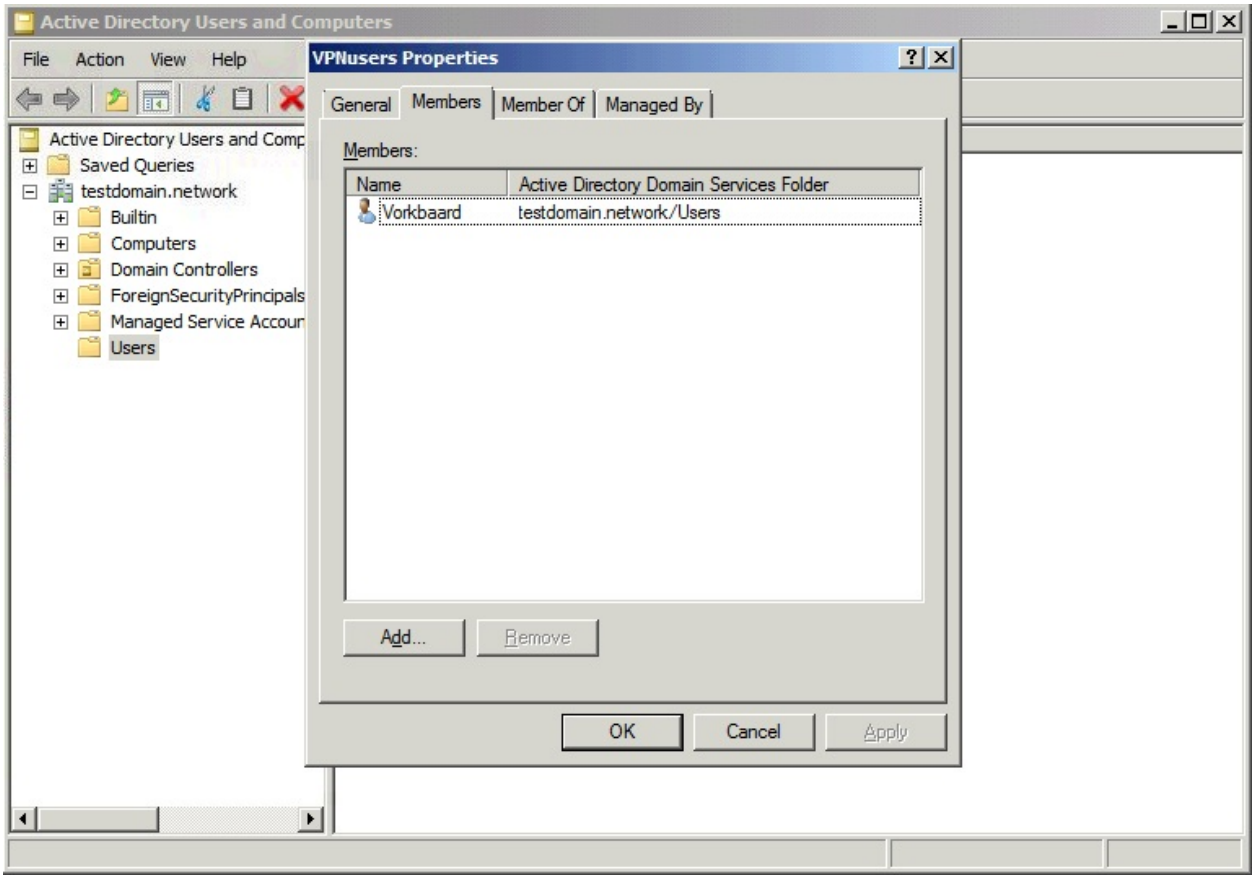
On the Active Directory domain controller

34.51.4 Create a group VPNusers

Create a security group in Active Directory Users and Computers called *VPNusers*. Everyone could have access but it's a good idea to keep some granular control over it.



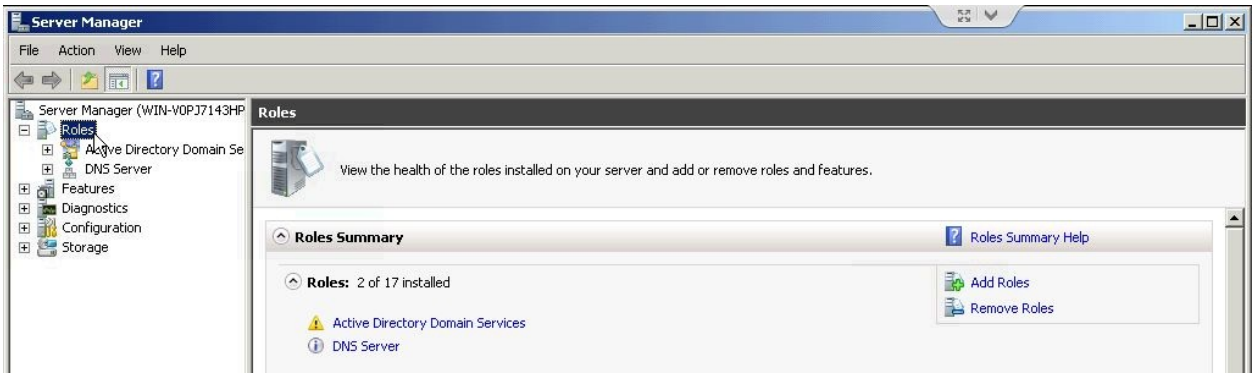
Add all accounts that need to use the VPN system to this group.



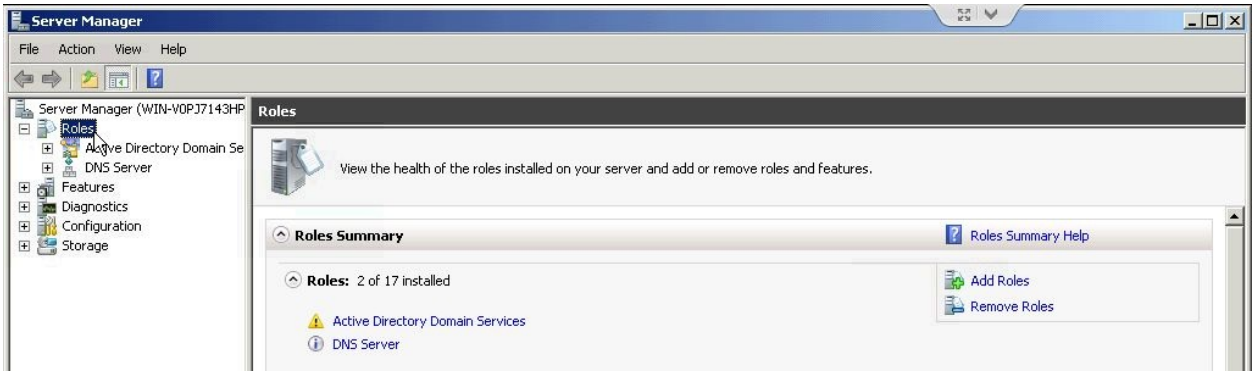
34.51.5 Install and configure RADIUS

If RADIUS isn't already set up, add the role to the Domain Controller. If it is set up, skip this step.

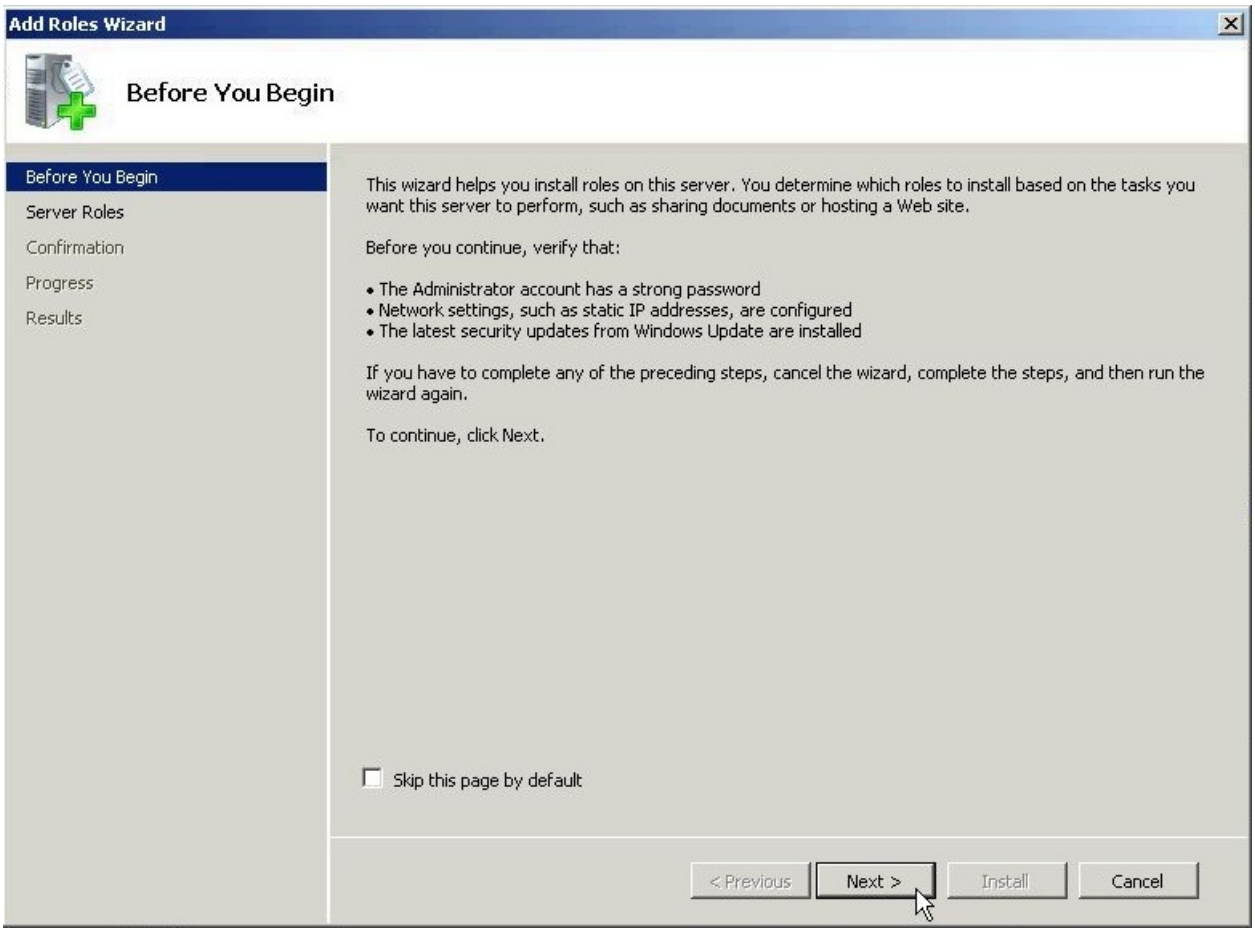
Open **Server Manager** and click the **Roles** node in the tree on the left.



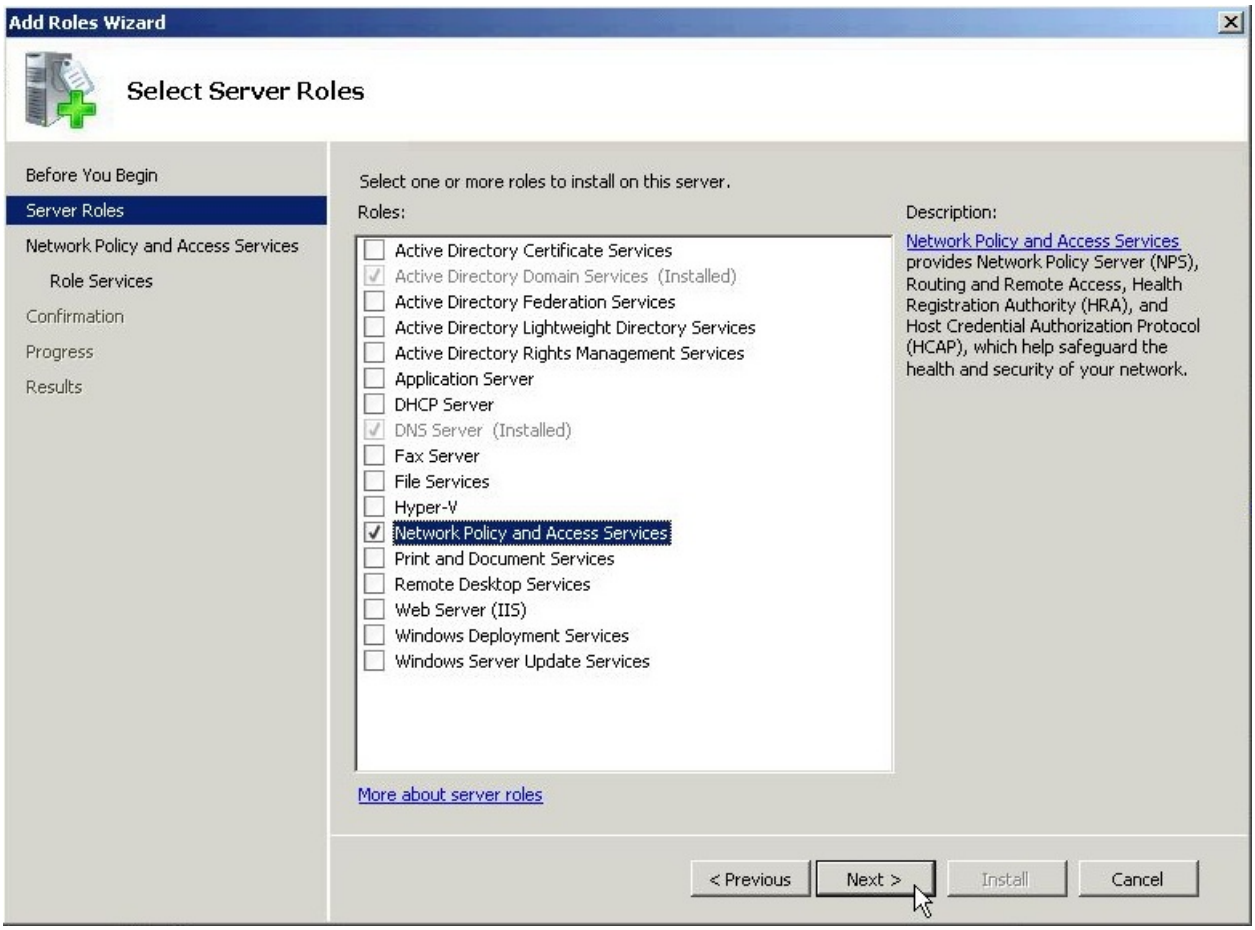
On the right side, click **Add Roles**.



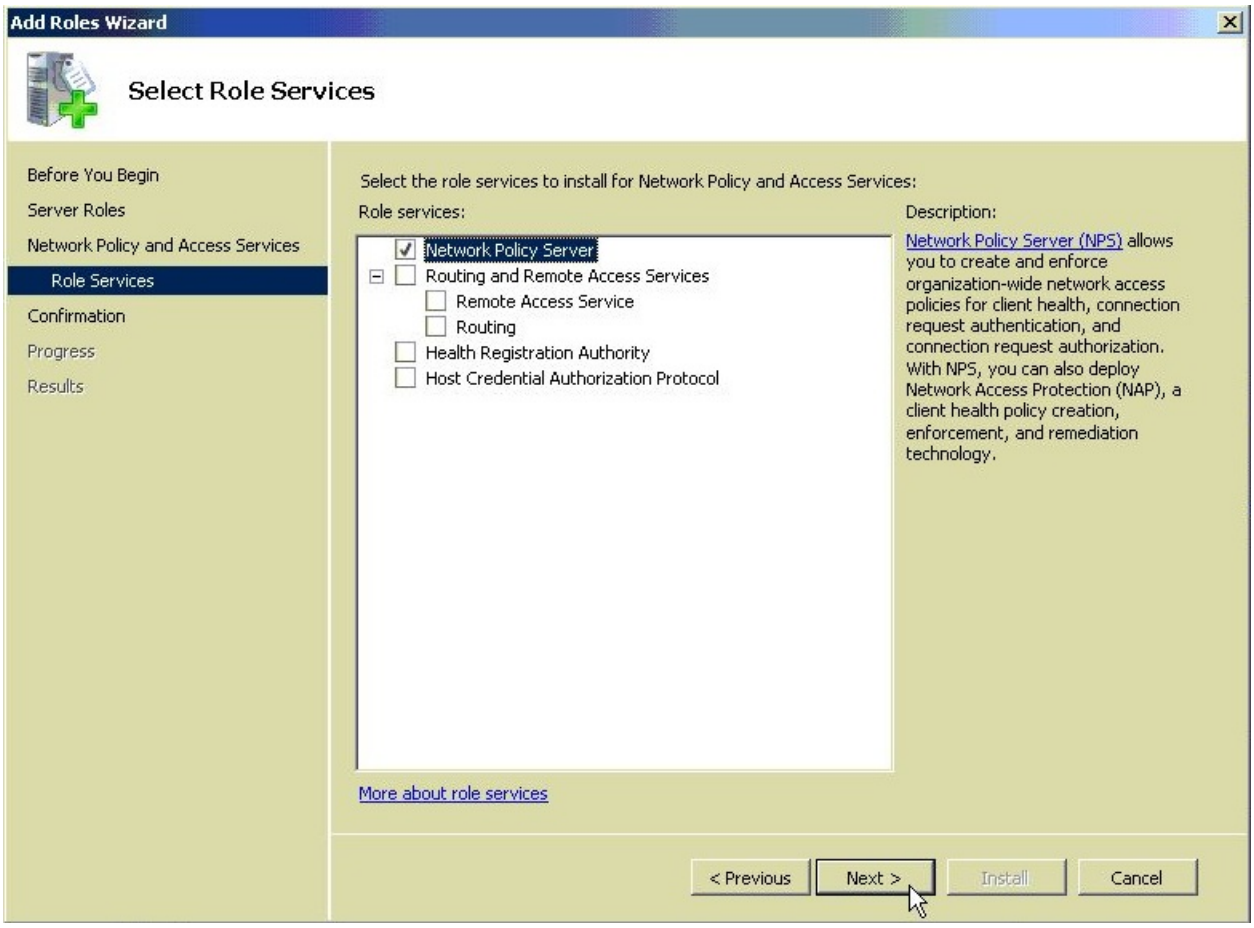
This will open the Add Roles Wizard.



Check **Network Policy and Access Services**.



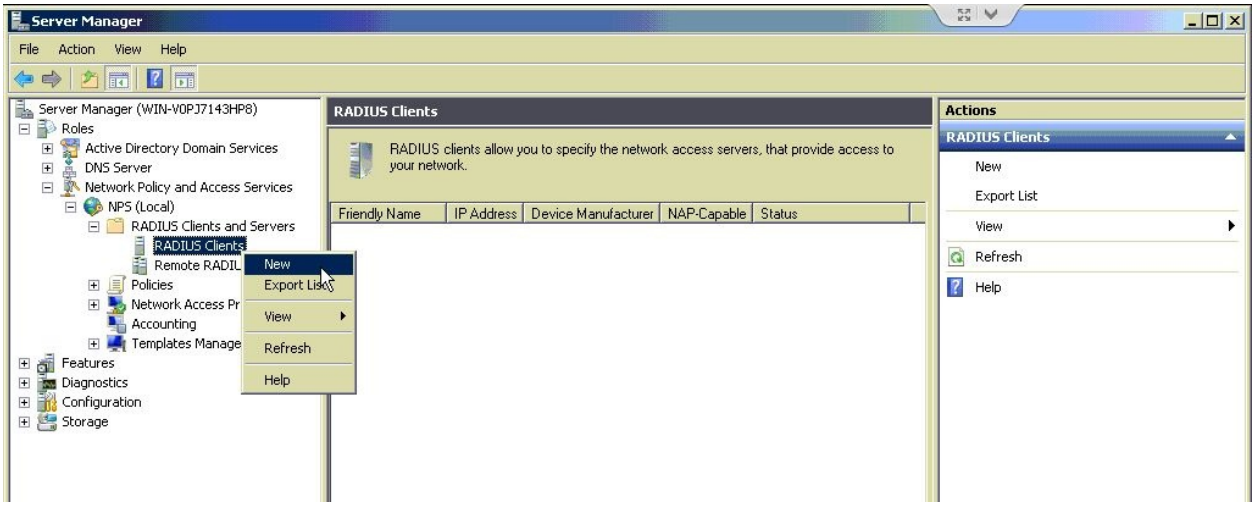
Select **Network Policy Server**.



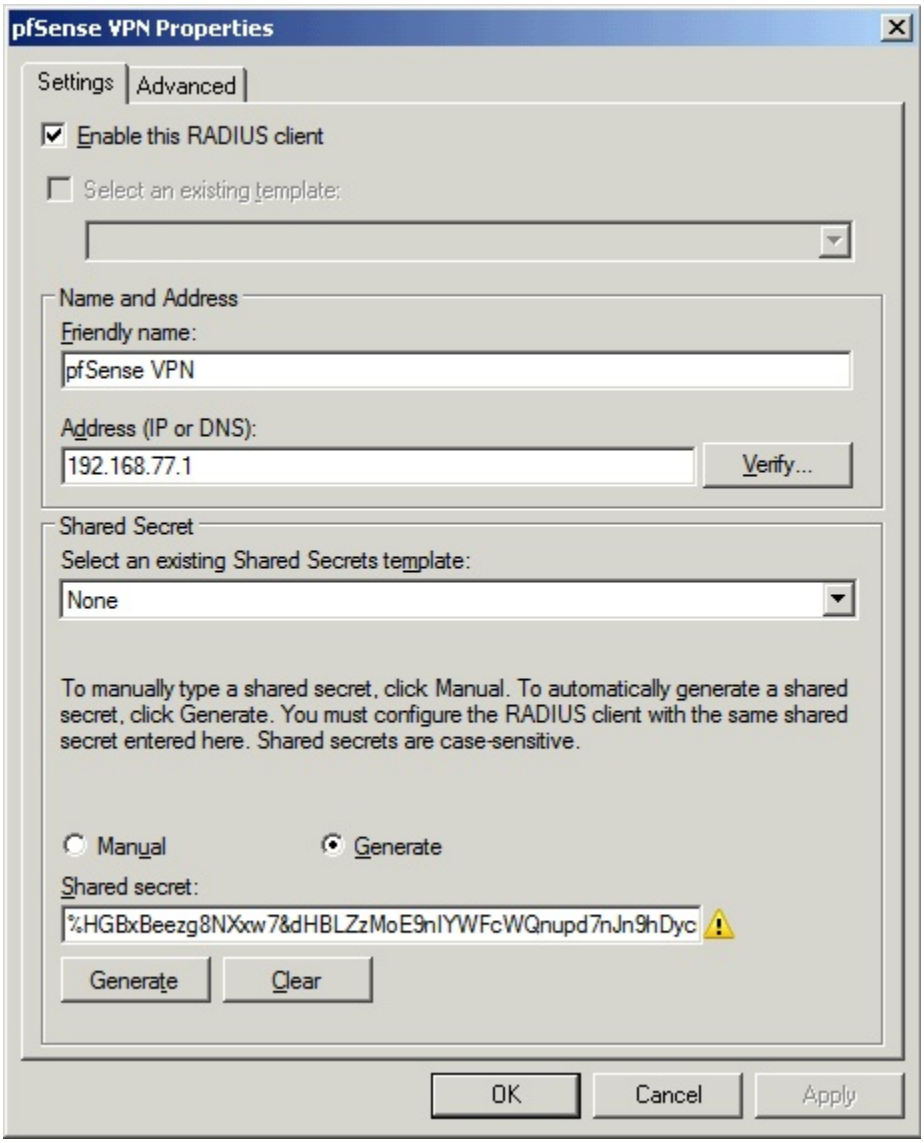
If all went well there is now a ***Network Policy and Access Services*** node in the tree.



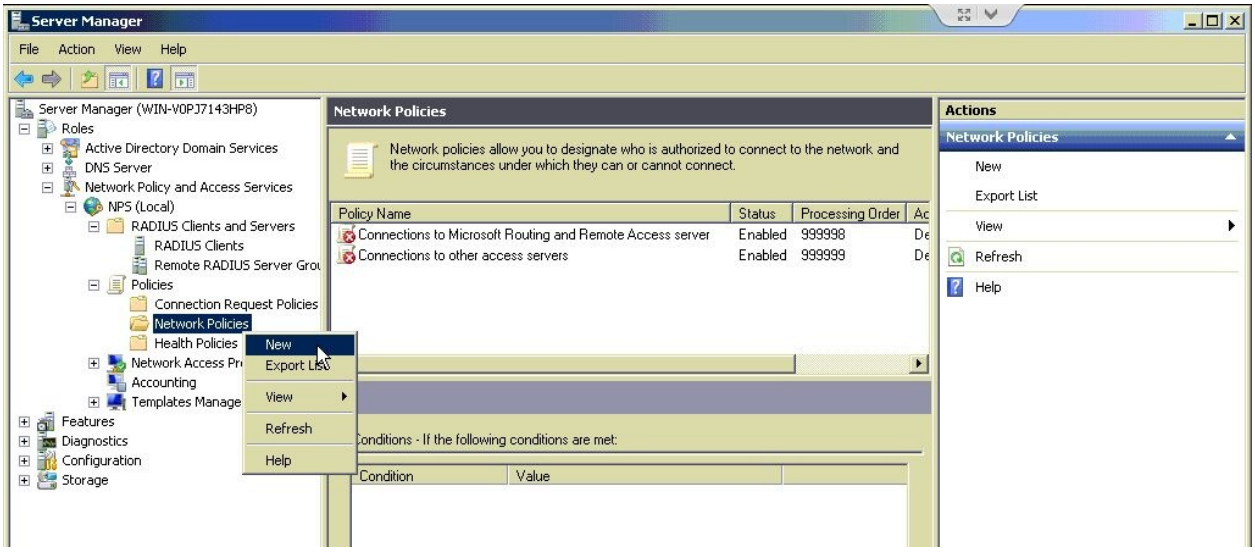
Expand the **Network Policy and Access Services** node, go to **NPS (Local) > RADIUS Clients and Servers**, right-click **RADIUS Clients** and choose **New**.



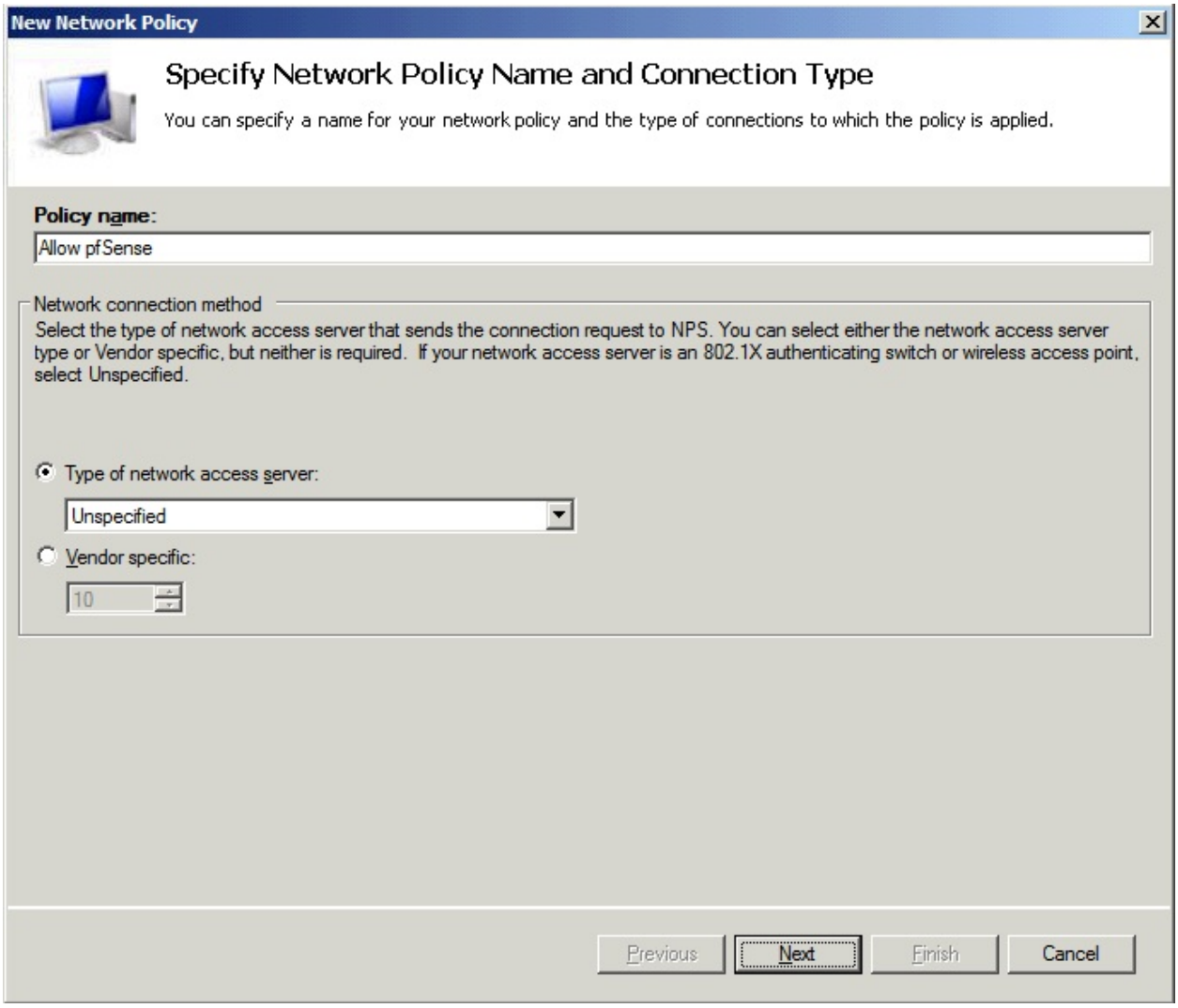
In the **Friendly name** field, enter **pfSense VPN** or anything deemed appropriate. In the **Address (IP or DNS)** field, enter the IP address of the pfSense firewall. Mine is *192.168.77.1*. **Shared Secret**: check **Generate** and save the shared secret; It will be needed later on.



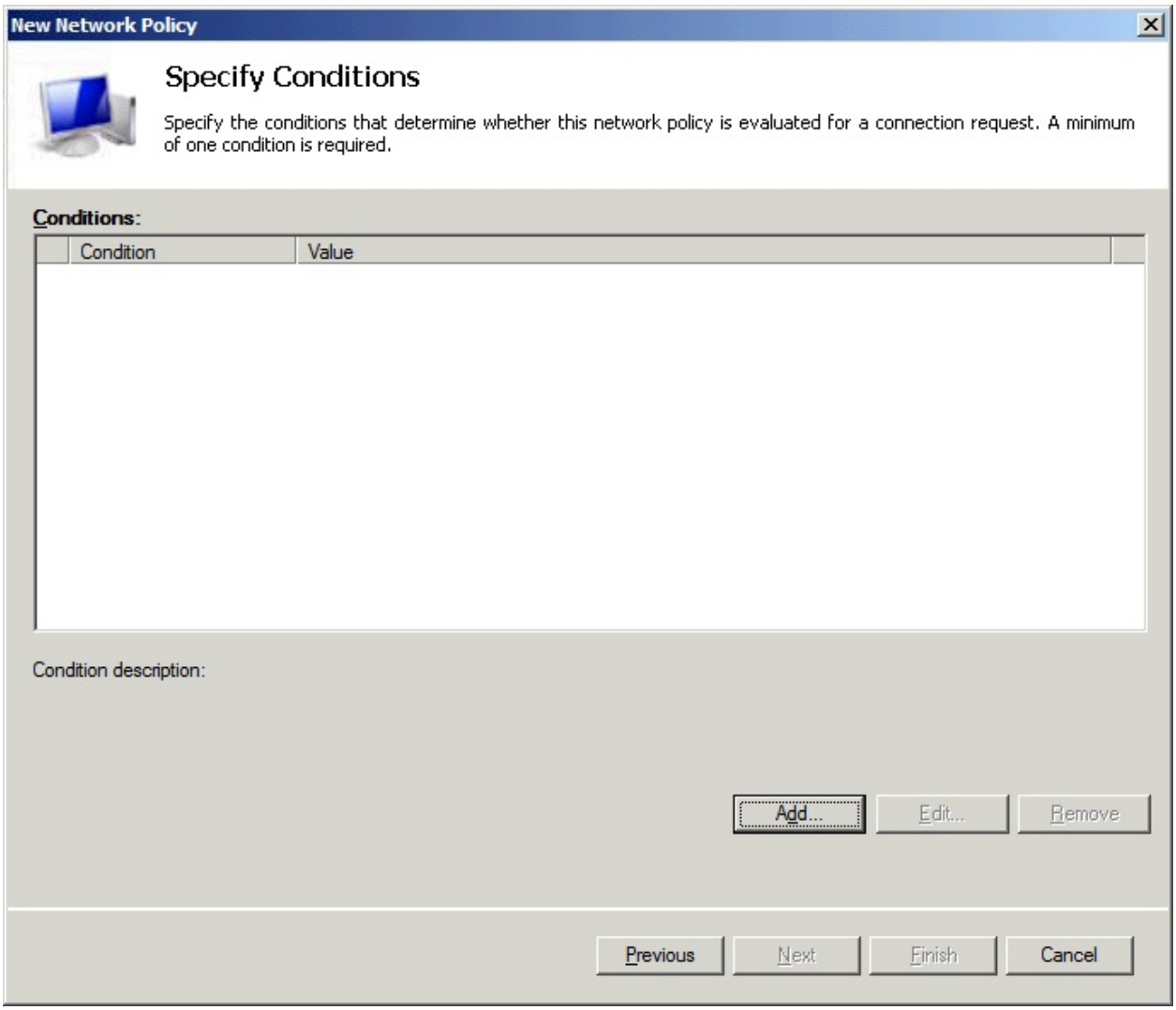
Under **NPS (Local) > Policies** right-click **Network Policies** and select **New**.



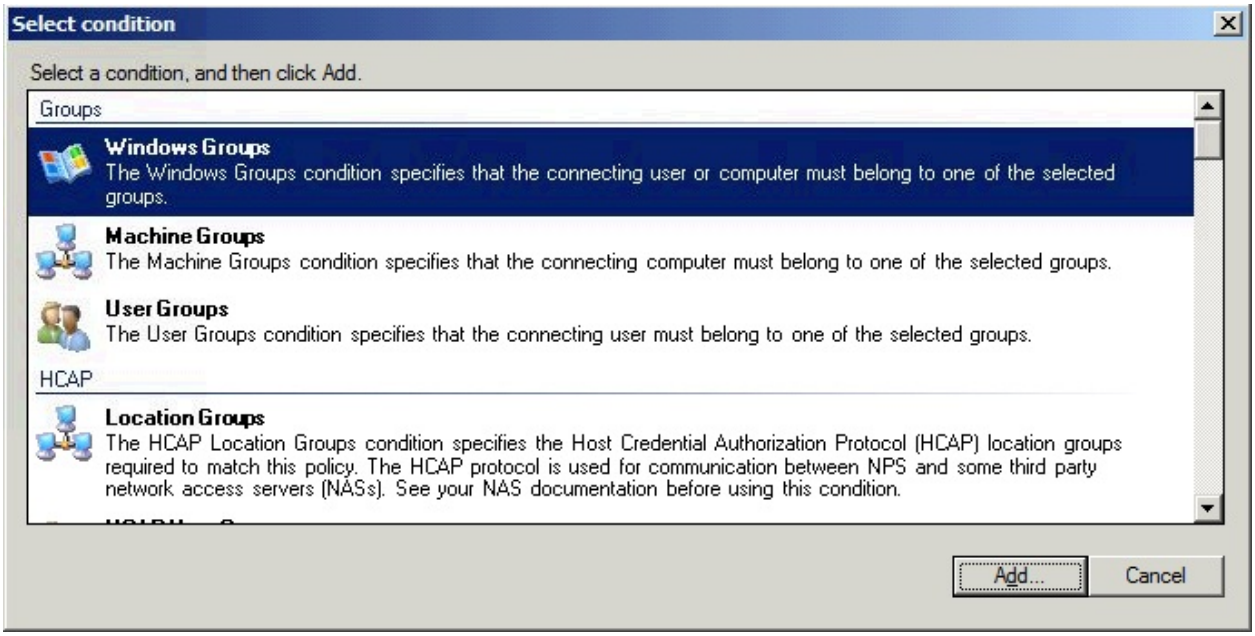
In the **Policy name** field, enter **Allow pfSense**. **Type of network access server**: **Unspecified**.



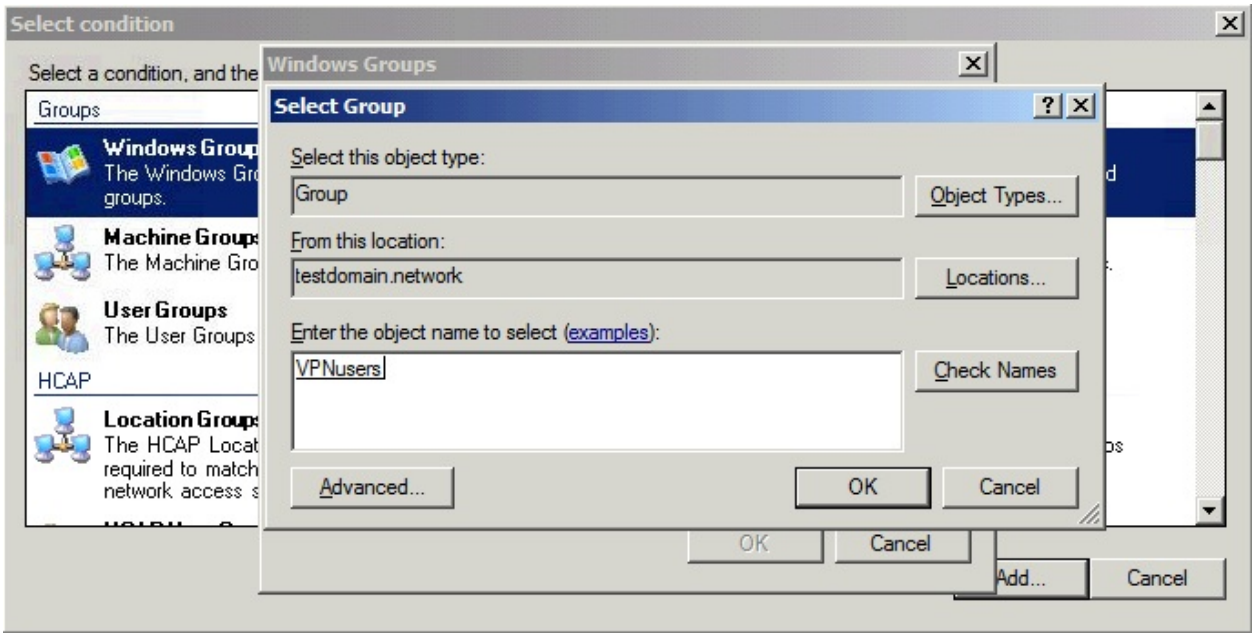
In the **Specify Conditions** window, click **Add...**



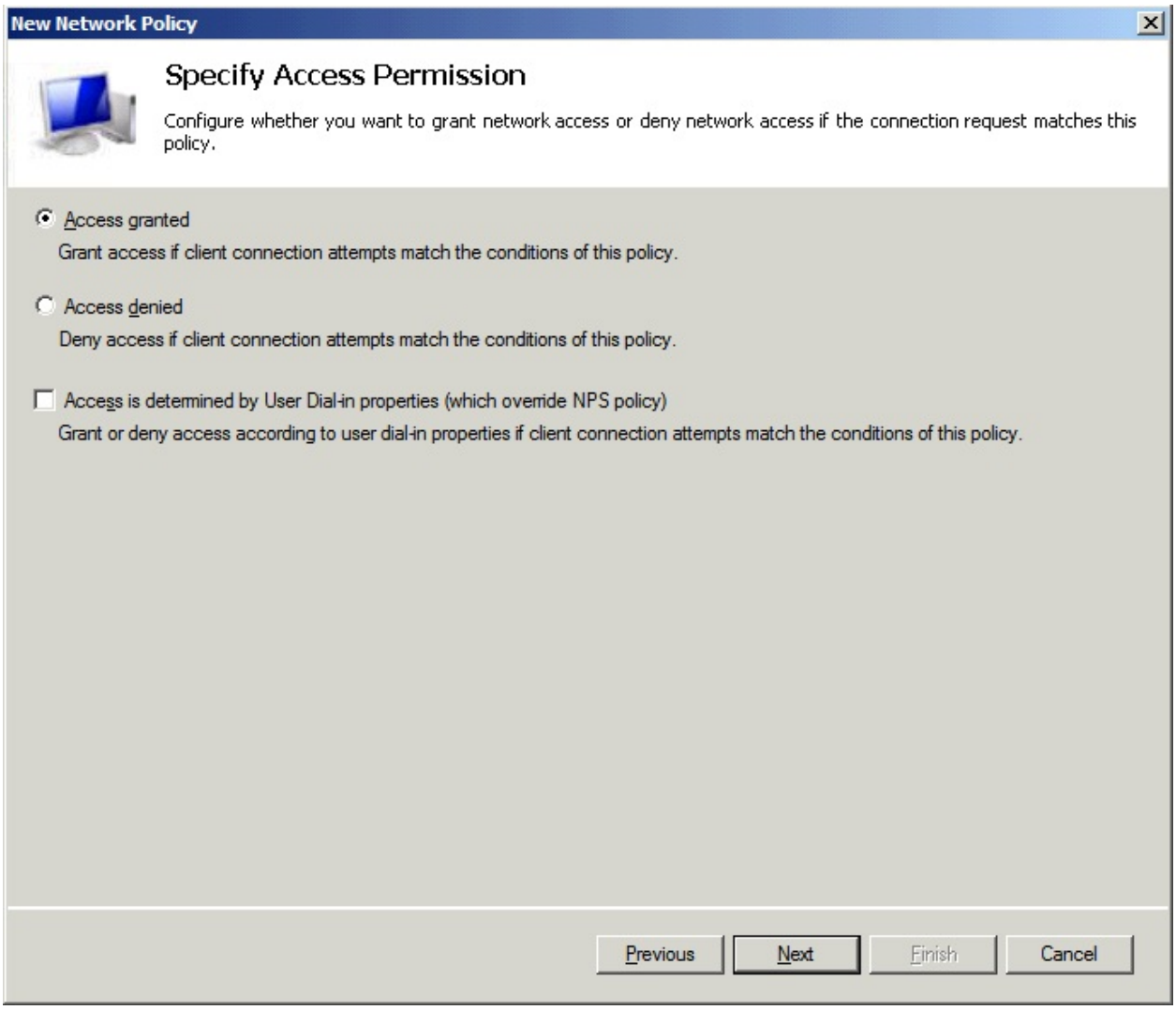
Select **Windows Groups** and click **Add...**



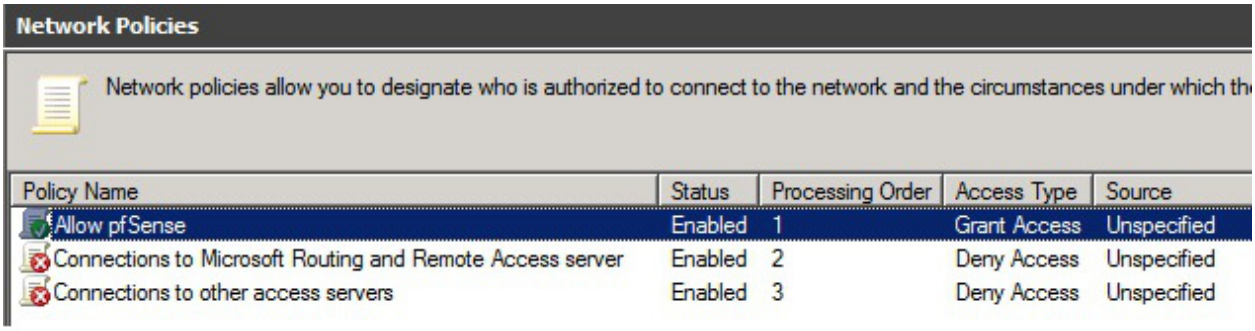
Click **Add Groups…** and add the group **VPNusers** (or whatever group is needed).



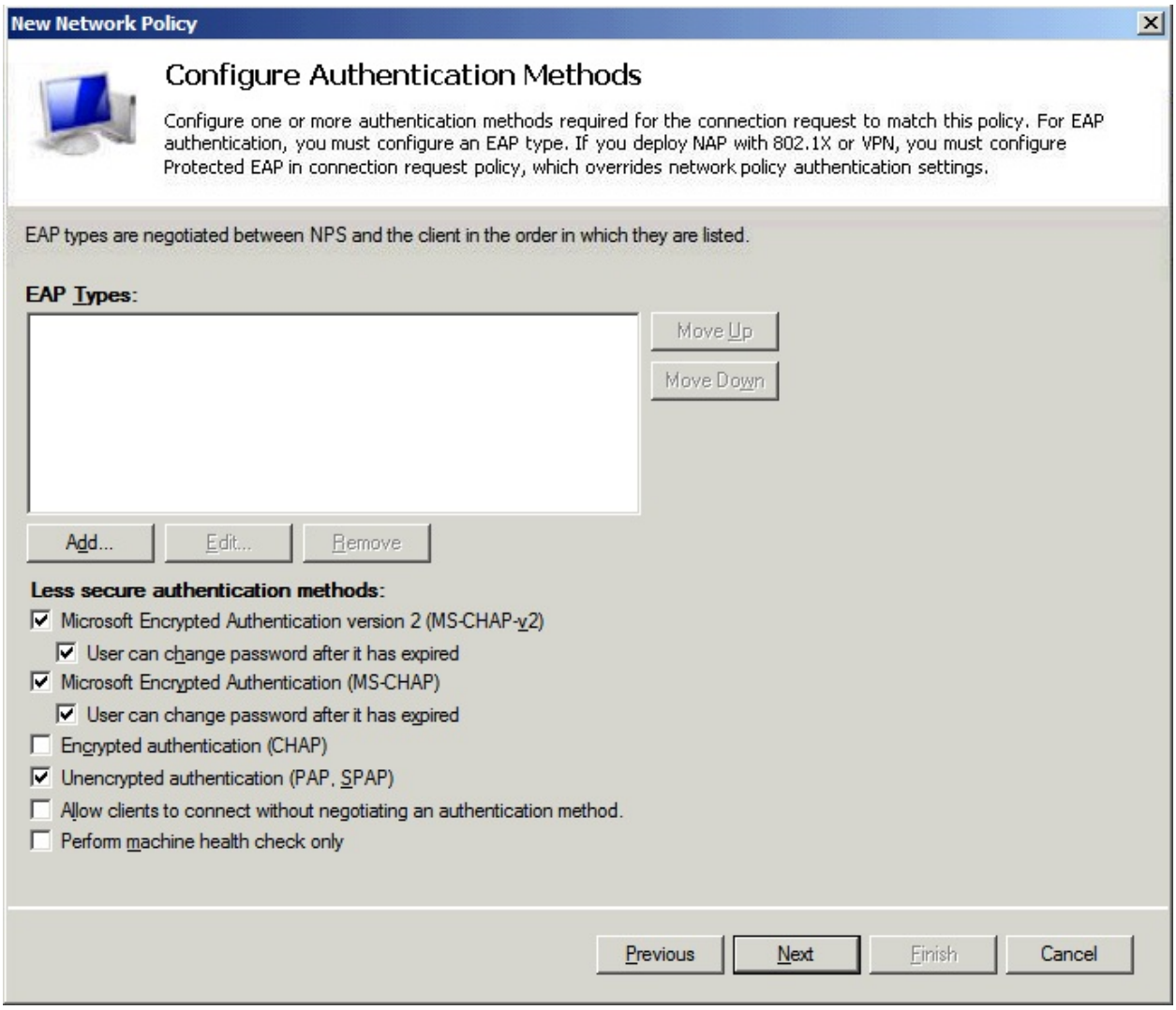
Back in the **Specify Conditions** window, click **Next** and select **Access granted**.



Put the new policy before policies preventing the connection. Mind the **Processing Order** field. Thanks to Dan for alerting me on this.



In the **Configure Authentication Methods** window, check **Unencrypted authentication (PAP, SPAP)**.



Skip the next wizard window (**Constraints**) or configure it if desired. I suggest leaving it as it is until after confirming the VPN works.

It's done. Next, Next, Finish until the end.

On the pfSense firewall

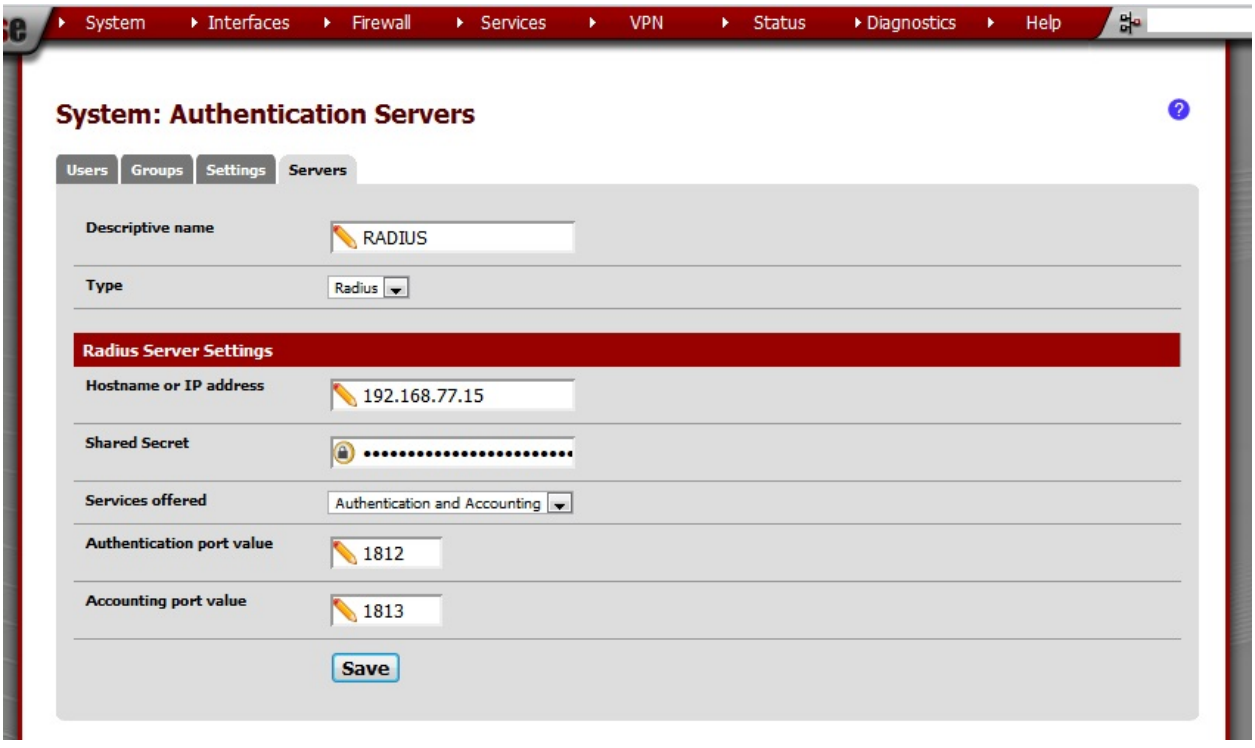
34.51.6 Set up the Authentication Server

In the pfSense webGUI, go to **System > User Manager**, on the **Servers** tab. Click  on the right.



Enter these values:

Descriptive name	RADIUS
Type	Radius
Hostname or IP address	192.168.77.15
Shared Secret	Paste the shared secret generated by the RADIUS server. Then delete the file containing the shared secret. It will not be needed again and if it is, a new one may be generated instead.
Services offered	Authentication and Accounting
Authentication port value	1812
Accounting port value	1813



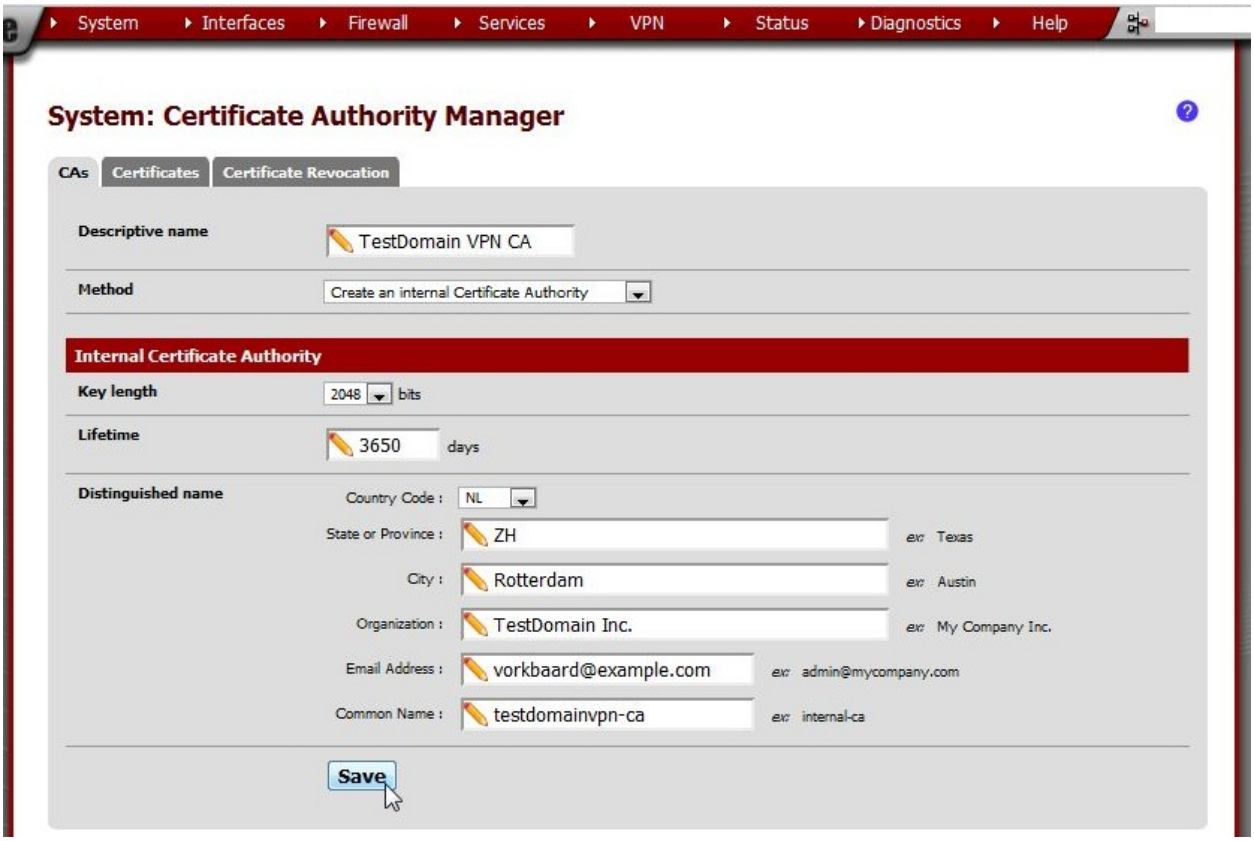
34.51.7 Install a Certificate Authority

Go to **System > Cert Manager**, **CAs** tab and click  .



Enter these values:

Descriptive name	TestDomain VPN CA
Method	Create an internal Certificate Authority
Key length	2048
Lifetime	3650 days Ten years should be enough for now.
Distinguished name	*Fill out the preferences here.*
Common name	testdomainvpn-ca

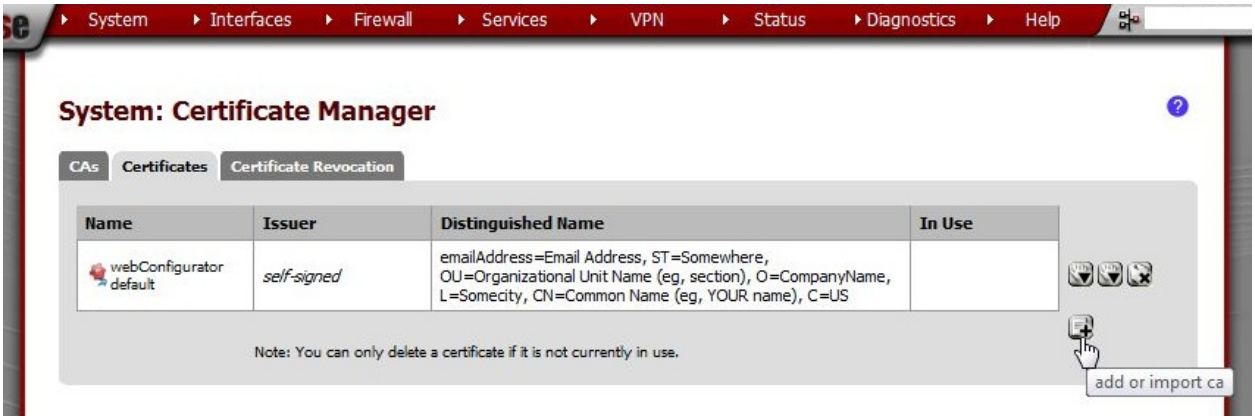


Note that now there is an extra CA in the CA list.



34.51.8 Create an internal certificate

Go to **System > Cert Manager**, **Certificates** tab and click  .



Enter these values:

Method	Create an internal Certificate
Descriptive name	vpn-testdomain-network
Certificate Authority	TestDomain VPN CA
Key length	2048
Certificate Type	Server Certificate
Lifetime	3560 days
Distinguished name	*Fill out the preferences here.*
Common Name	vpn.example.com

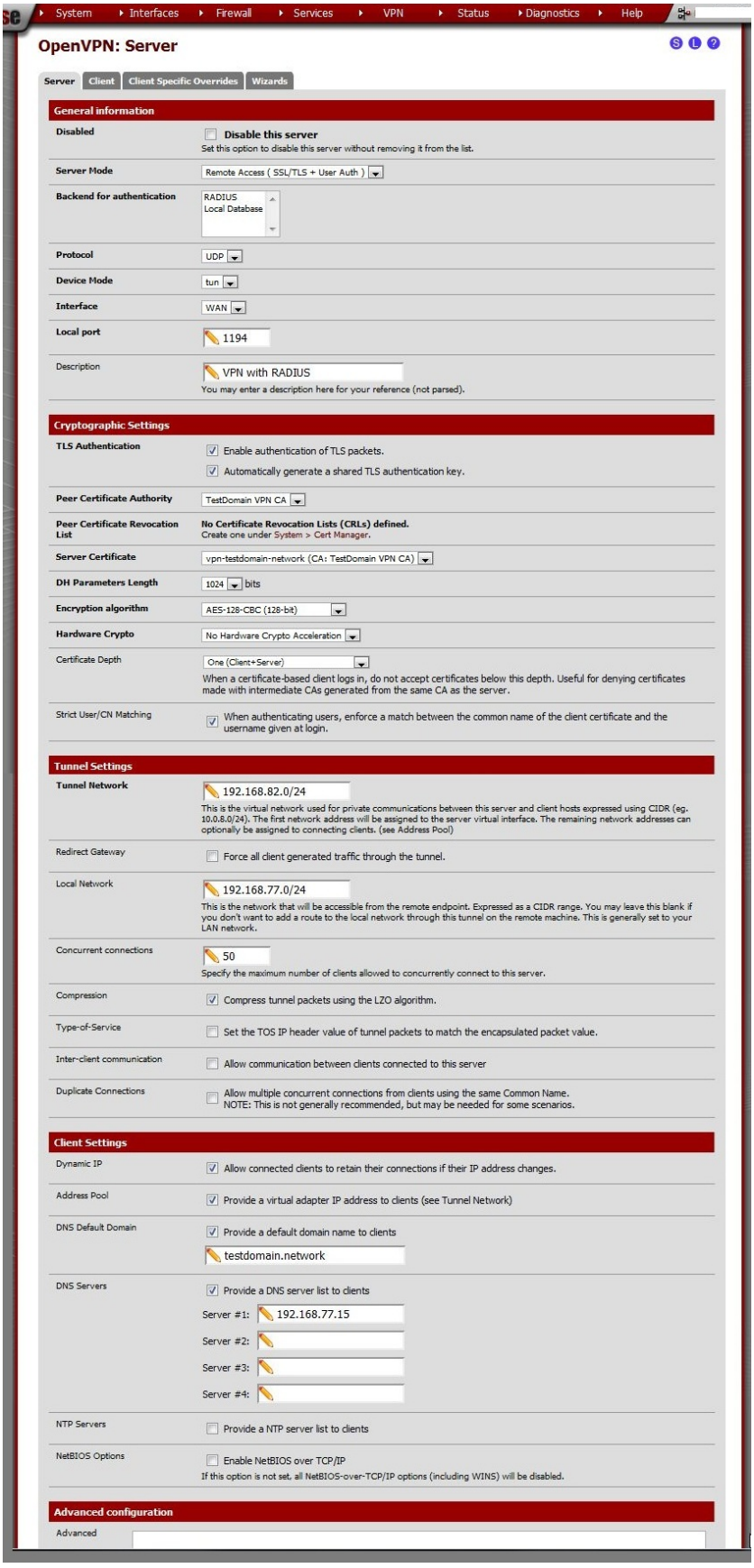
34.51.9 Set up the OpenVPN server

Go to **VPN > OpenVPN**, **Servers** tab and click  .



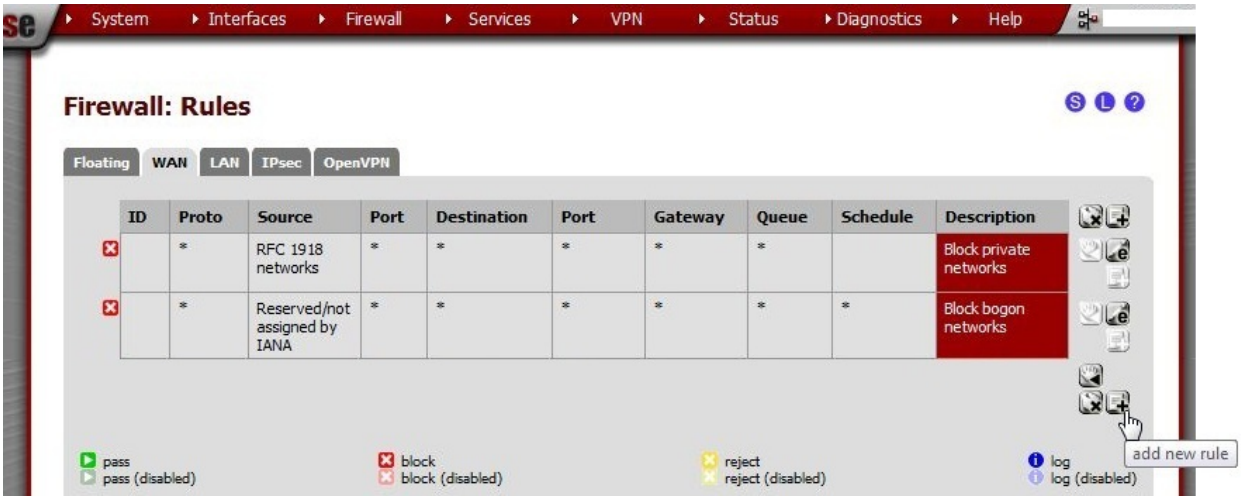
Enter these values:

Server Mode:	Remote Access (SSL/TLS User Auth)
Backend for authentication	RADIUS
Protocol	UDP
Device Mode	tun
Interface	WAN
Local port	1194
Description	Something appropriate
TLS Authentication	Check both **Enable authentication of TLS packets** and **Automatically generate a shared TLS authentication key**.
Peer Certificate Authority	TestDomain VPN CA
Server Certificate	vpn-testdomain-network (CA: TestDomain VPN CA)
DH Parameters Length	1024
Encryption algorithm	AES-128-CBC (128-bit) Others probably work as well.
Hardware Crypto	No Hardware Crypto Acceleration
Certificate Depth	One (Client Server)
Strict User/CN Matching	If this is checked, a user can only connect with their own credentials, not that of other users. I think this is is good idea, so check this option.
Tunnel Network	192.168.82.0/24 Or any other network, as long as it is not in use in the LAN/WAN and probably not at users' locations. i.e. don't use *192.168.0.0/24*, *192.168.1.0/24* and *10.0.0.0/24*.
Redirect Gateway	If this is checked, not only traffic to the LAN will be routed through the tunnel but also to the rest of the Internet. If the user starts downloading a movie it will go through the company network. On the other hand, they will be behind the corporate firewall. Check this to use the VPN for secure Internet access. Do not check if the corporate line has a slow upload speed.
Local Network	192.168.77.0/24 This is my range. Enter the actual LAN subnet here.
Concurrent connections	Crypto can be tough on resources. If the pfSense installation runs on an appliance keep this number low. If it runs on an old computer it can do more. Keep en eye on the machine's CPU. If more concurrent VPN connections ask too much of resources, upgrade the hardware. I tend to set this number to the number of client installations.
Compression	Check, unless clients and server are on stone-age hardware.
Type-of-Service	Unchecked
Inter-client communication	Unchecked unless needed.
Duplicate Connections	Unchecked unless needed.
Dynamic IP	Checked unless seriously worried about laptops getting stolen in the middle of a VPN session or client connections being hijacked.
Address Pool	Checked


System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
OpenVPN: Server
Server
Client
Client Specific Overrides
Wizards
General information
Disabled
Disable this server
Set this option to disable this server without removing it from the list.
Server Mode
Remote Access (SSL/TLS + User Auth)
Backend for authentication
RADIUS
Local Database
Protocol
UDP
Device Mode
tun
Interface
WAN
Local port
1194
Description
VPN with RADIUS
You may enter a description here for your reference (not parsed).
Cryptographic Settings
TLS Authentication
Enable authentication of TLS packets.
Automatically generate a shared TLS authentication key.
Peer Certificate Authority
TestDomain VPN CA
Peer Certificate Revocation List
No Certificate Revocation Lists (CRLs) defined.
Create one under System > Cert Manager.
Server Certificate
vpn-testdomain-network (CA: TestDomain VPN CA)
DH Parameters Length
1024
bits
Encryption algorithm
AES-128-CBC (128-bit)
Hardware Crypto
No Hardware Crypto Acceleration
Certificate Depth
One (Client+Server)
When a certificate-based client logs in, do not accept certificates below this depth. Useful for denying certificates made with intermediate CAs generated from the same CA as the server.
Strict User/CN Matching
When authenticating users, enforce a match between the common name of the client certificate and the username given at login.
Tunnel Settings
Tunnel Network
192.168.82.0/24
This is the virtual network used for private communications between this server and client hosts expressed using CIDR (eg. 10.0.8.0/24). The first network address will be assigned to the server virtual interface. The remaining network addresses can optionally be assigned to connecting clients. (see Address Pool)
Redirect Gateway
Force all client generated traffic through the tunnel.
Local Network
192.168.77.0/24
This is the network that will be accessible from the remote endpoint. Expressed as a CIDR range. You may leave this blank if you don't want to add a route to the local network through this tunnel on the remote machine. This is generally set to your LAN network.
Concurrent connections
50
Specify the maximum number of clients allowed to concurrently connect to this server.
Compression
Compress tunnel packets using the LZO algorithm.
Type-of-Service
Set the TOS IP header value of tunnel packets to match the encapsulated packet value.
Inter-client communication
Allow communication between clients connected to this server
Duplicate Connections
Allow multiple concurrent connections from clients using the same Common Name.
NOTE: This is not generally recommended, but may be needed for some scenarios.
Client Settings
Dynamic IP
Allow connected clients to retain their connections if their IP address changes.
Address Pool
Provide a virtual adapter IP address to clients (see Tunnel Network)
DNS Default Domain
Provide a default domain name to clients
testdomain.network
DNS Servers
Provide a DNS server list to clients
Server #1:
192.168.77.15
Server #2:
Server #3:
Server #4:
NTP Servers
Provide a NTP server list to clients
NetBIOS Options
Enable NetBIOS over TCP/IP
If this option is not set, all NetBIOS-over-TCP/IP options (including WINS) will be disabled.
Advanced configuration
Advanced

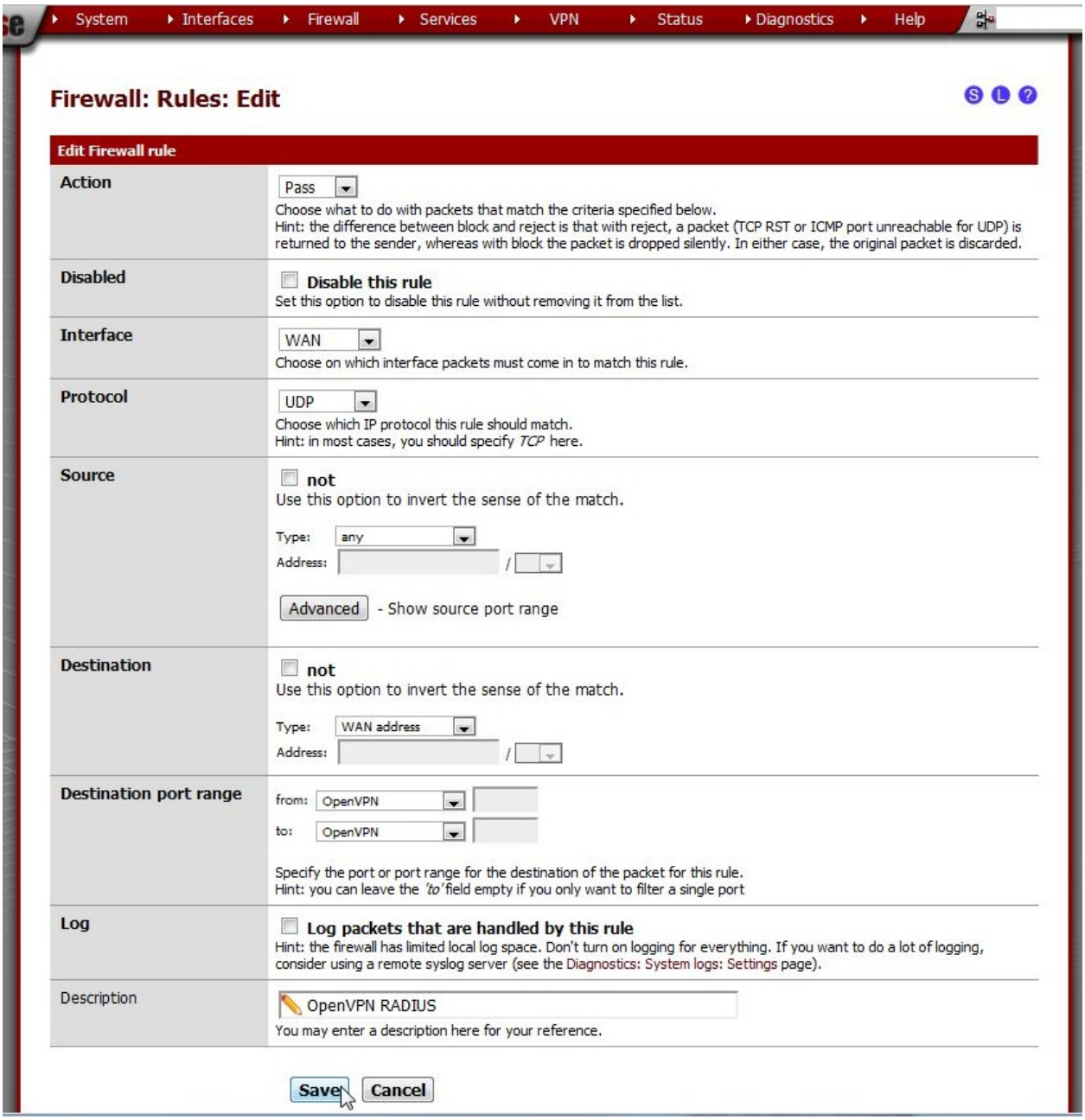
34.51.10 Configure the firewall

Go to **Firewall > Rules**, **WAN** tab and click  to create a new rule.



Enter these values:

Action	Pass
Disabled	not checked
Interface	WAN
Protocol	UDP
Source	unchecked, any
Destination	unchecked, WAN address
Destination port range	from OpenVPN to OpenVPN
Log	only check when troubleshooting
Description	OpenVPN RADIUS



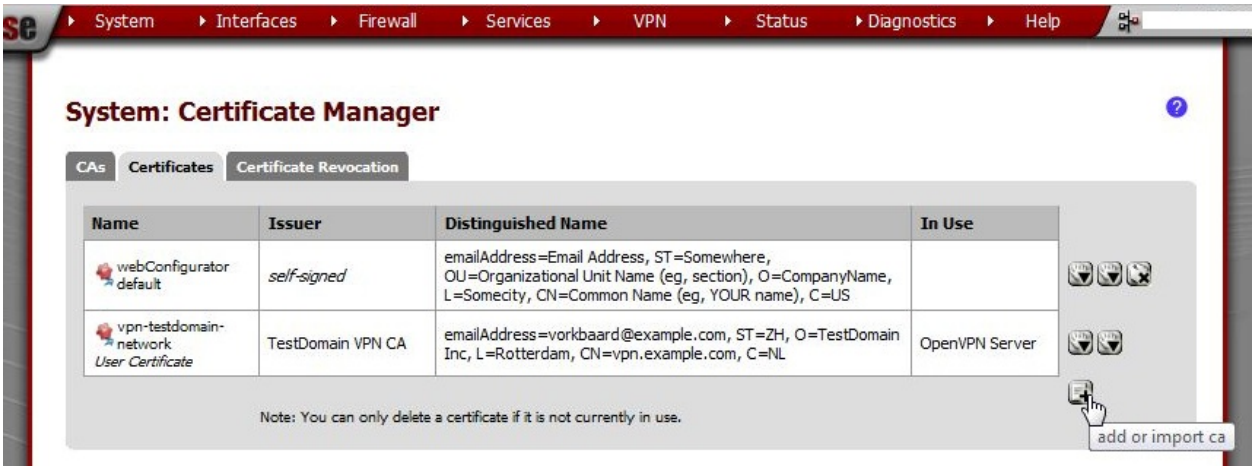
Click **Save** and the rules page will reload. Do not forget to click **Apply Changes**.



34.51.11 Create a Certificate

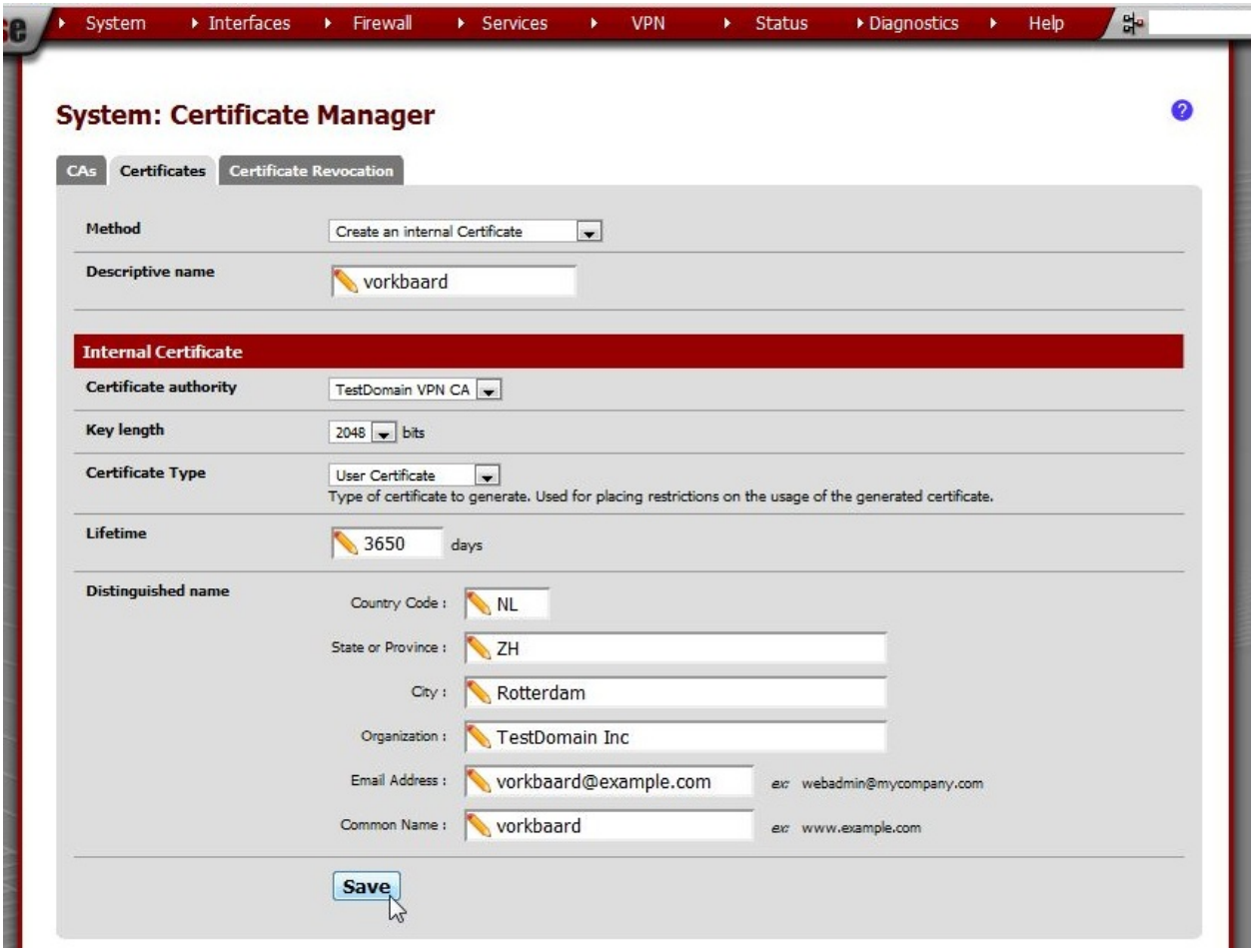
A certificate must be created for each user that is going to use the VPN system. In **Descriptive** and **Common Name**, enter the username the user uses to log on to Active Directory. Strictly speaking **Descriptive name** can be anything but usernames should be unique anyway.

Go to **System > Cert Manager** (not User Manager!), **Certificates** tab and click  .

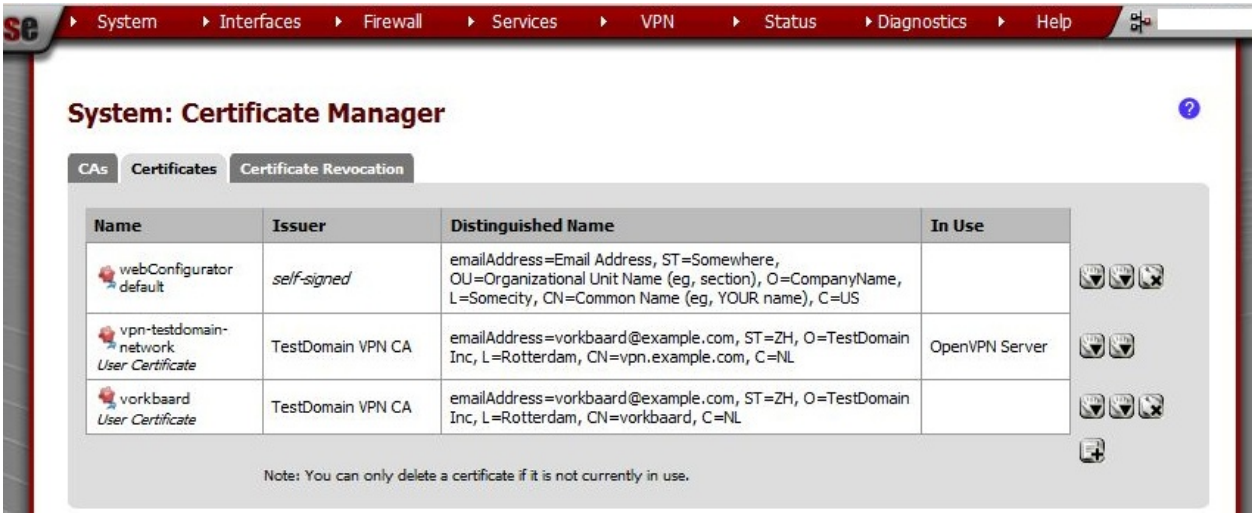


Enter these values:

Method	Create an internal Certificate
Descriptive name	[Username of the user that will be using the vpn connection] In some cases this is case sensitive. I tend to stick to all lowercase for that reason. It doesn't really matter but keep it in mind if the connection can't be established.
Certificate authority	TestDomain VPN CA
Key length	2048
Certificate Type	User Certificate
Lifetime	3650 days Unless the user has a temporary account.
Distinguished name	Fill out the preferences here.
Common Name:	[see Descriptive name]

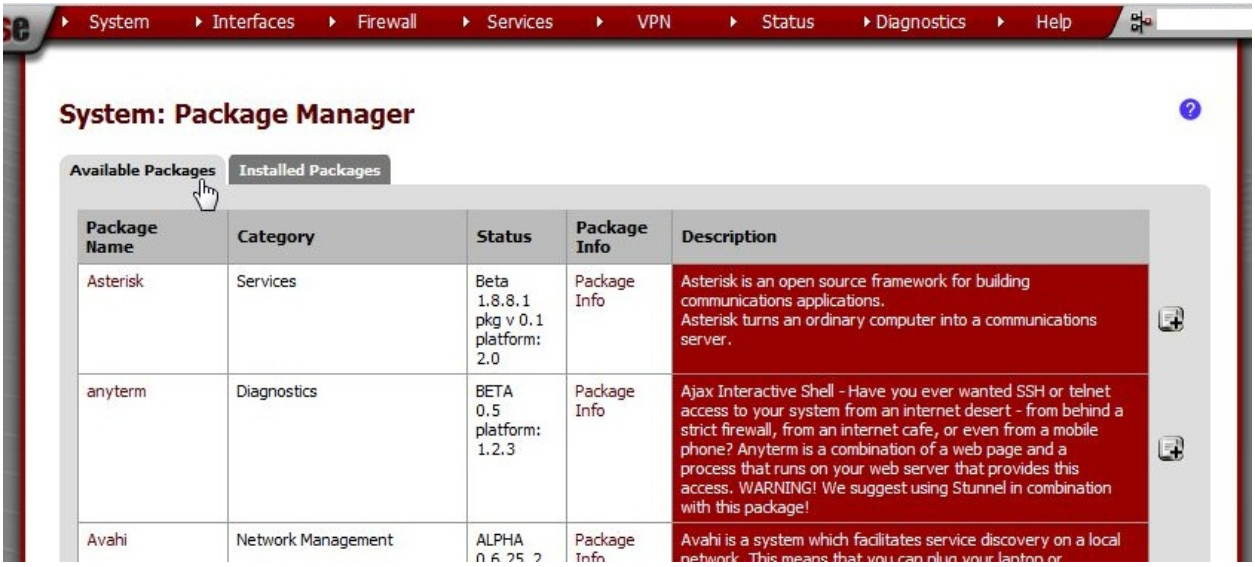


Note the entry in the Certificate list.

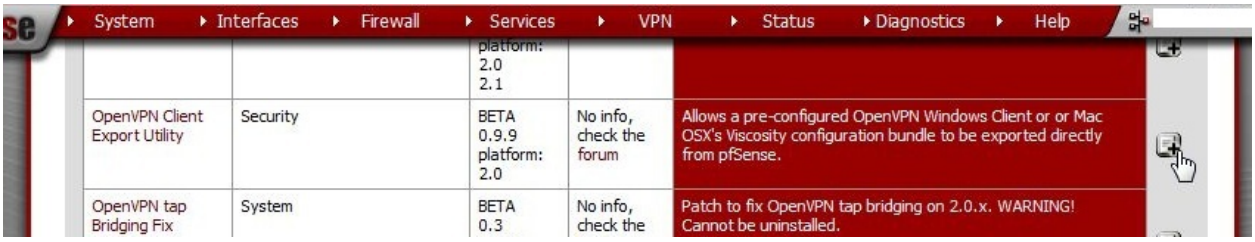


34.51.12 Install the OpenVPN Client Export Package

Go to **System > Packages**, **Available Packages** tab.

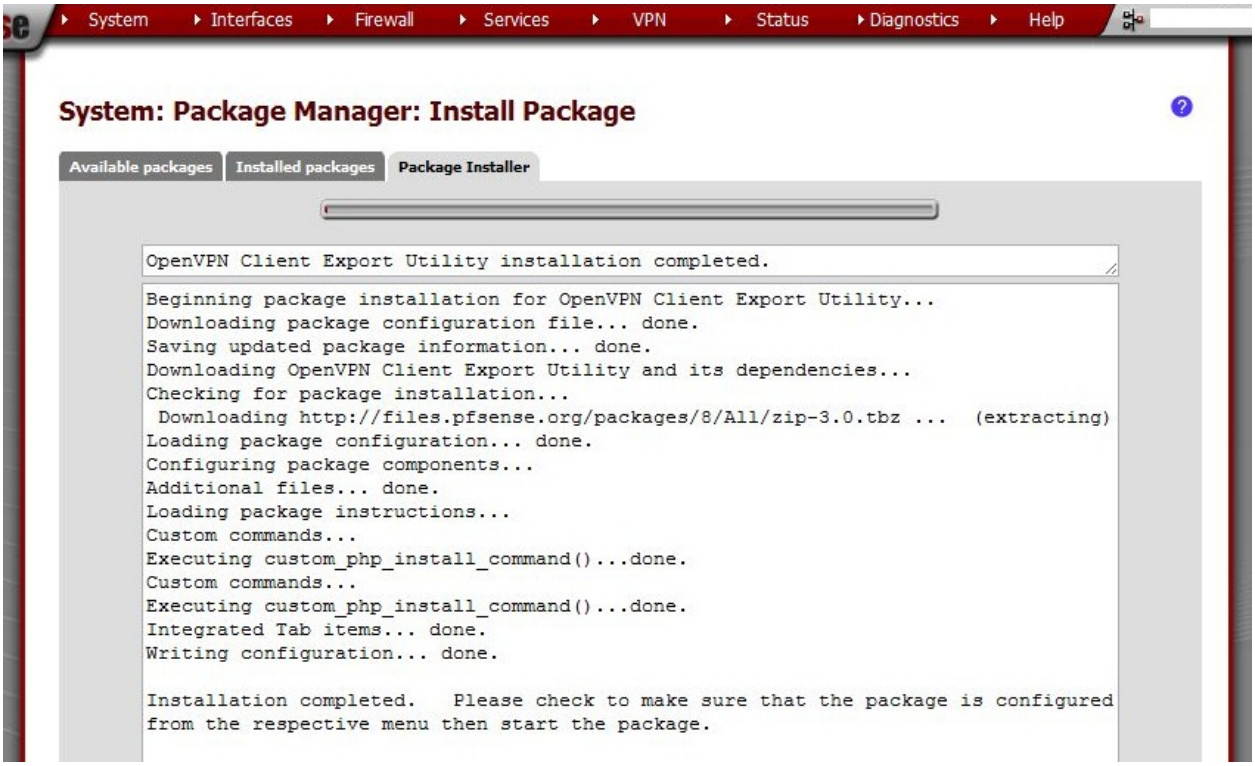


Scroll down to *OpenVPN Client Export Package* and click  on the right.



Confirm the selection and the package will be installed.

When it says **Installation completed** the installation is finished.



34.51.13 Prepare the Windows package

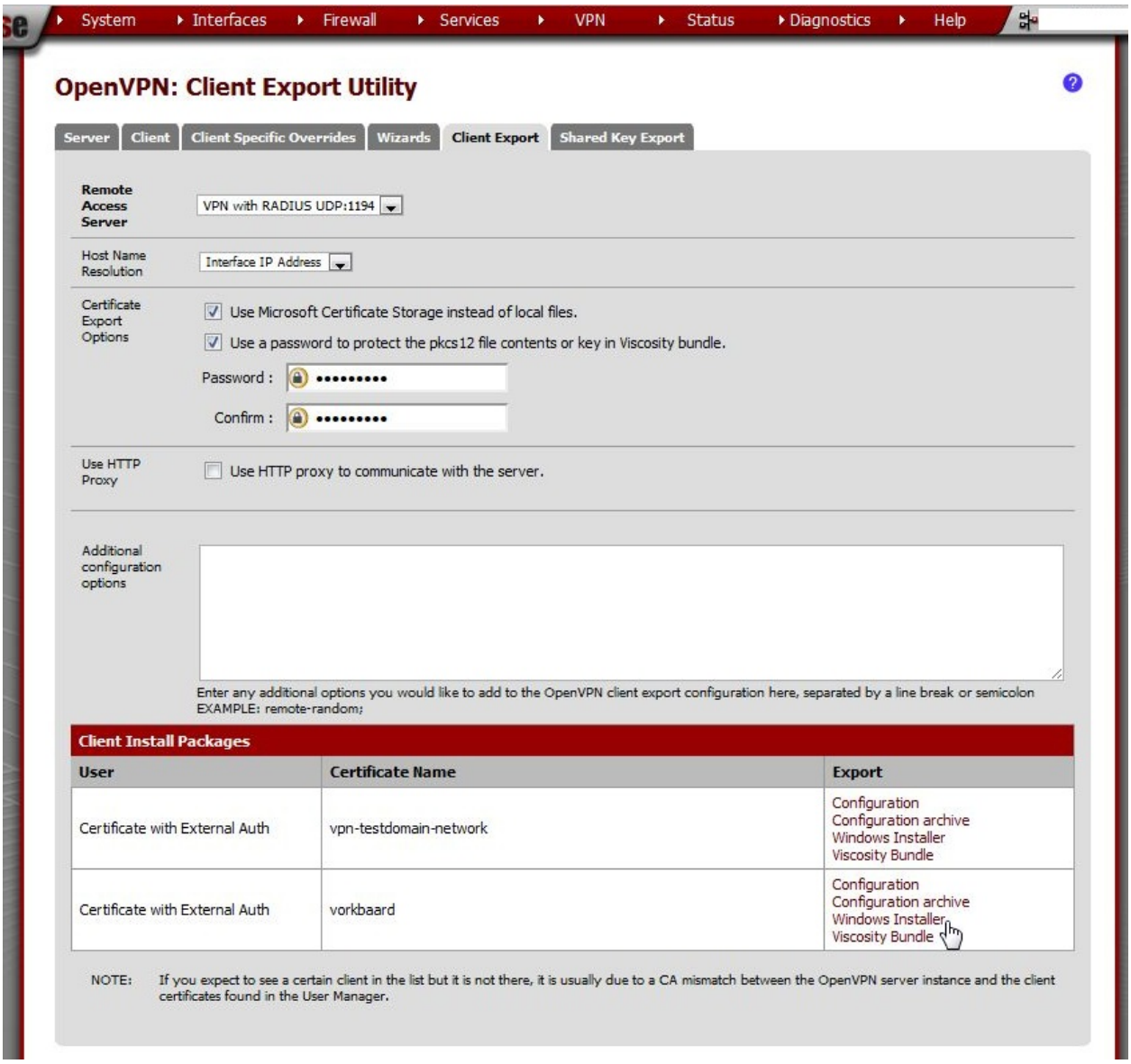
Go to **VPN > OpenVPN** and note that there is an extra tab called **Client Export**. Click it.



Enter these values:

Remote Access Server	VPN with RADIUS UDP:1194
Host Name Resolution	- If WAN has a static IP, enter *Interface IP Address* here. - If there is a DNS address pointing to the firewall, enter *Installation hostname* here. Personally, I like to create a dedicated DNS entry for VPN connections called *vpn.example.com*. If IP addresses / ISP connections are moved around it is nice to have things set up modularly. If unsure, stick with *Interface IP Address* for now.
Use Microsoft Certificate Storage instead of local files	checked
Use a password to protect the pkcs12 file contents or key in Viscosity bundle.	checked; choose a random password here and save it for use when installing the certificate on the client.
Use HTTP Proxy	Unchecked unless needed.

Find the right username under **Certificate Name** and then in the **Windows Installer** section, choose an appropriate installer for the user's platform, such as **x64-win6** for a 64-bit installer for Windows Vista and later.



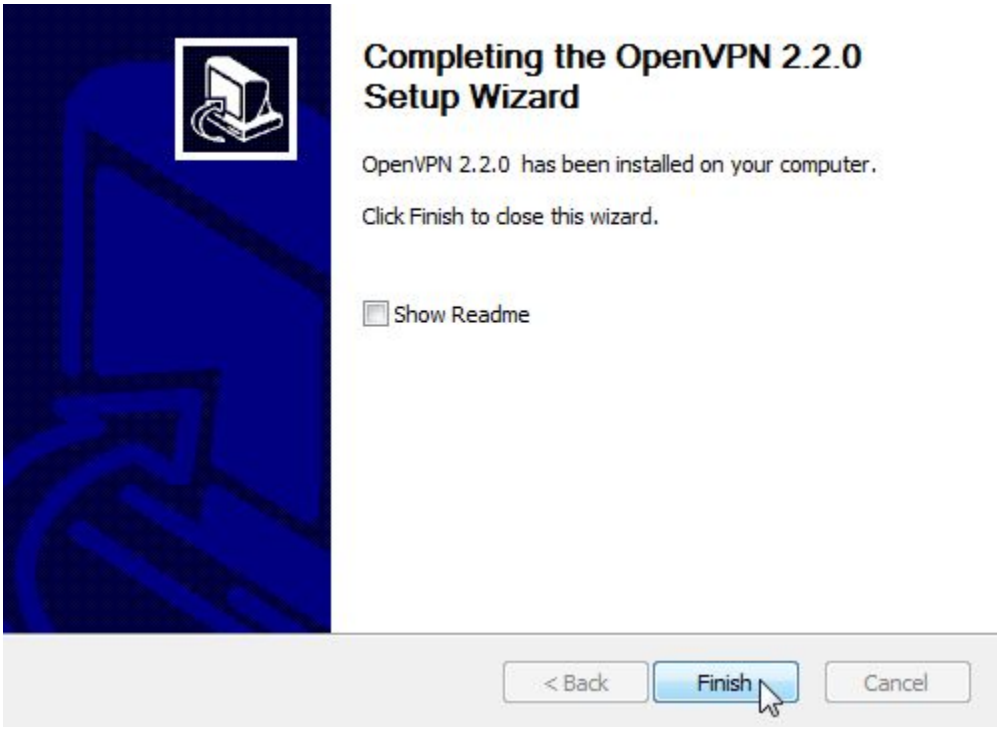
Get a package for each user.

On the Windows clients

34.51.14 install the OpenVPN package

Copy the downloaded Windows Installed to the client. It is named after the tunnel configuration, for example *router-udp-1194-install.exe*.

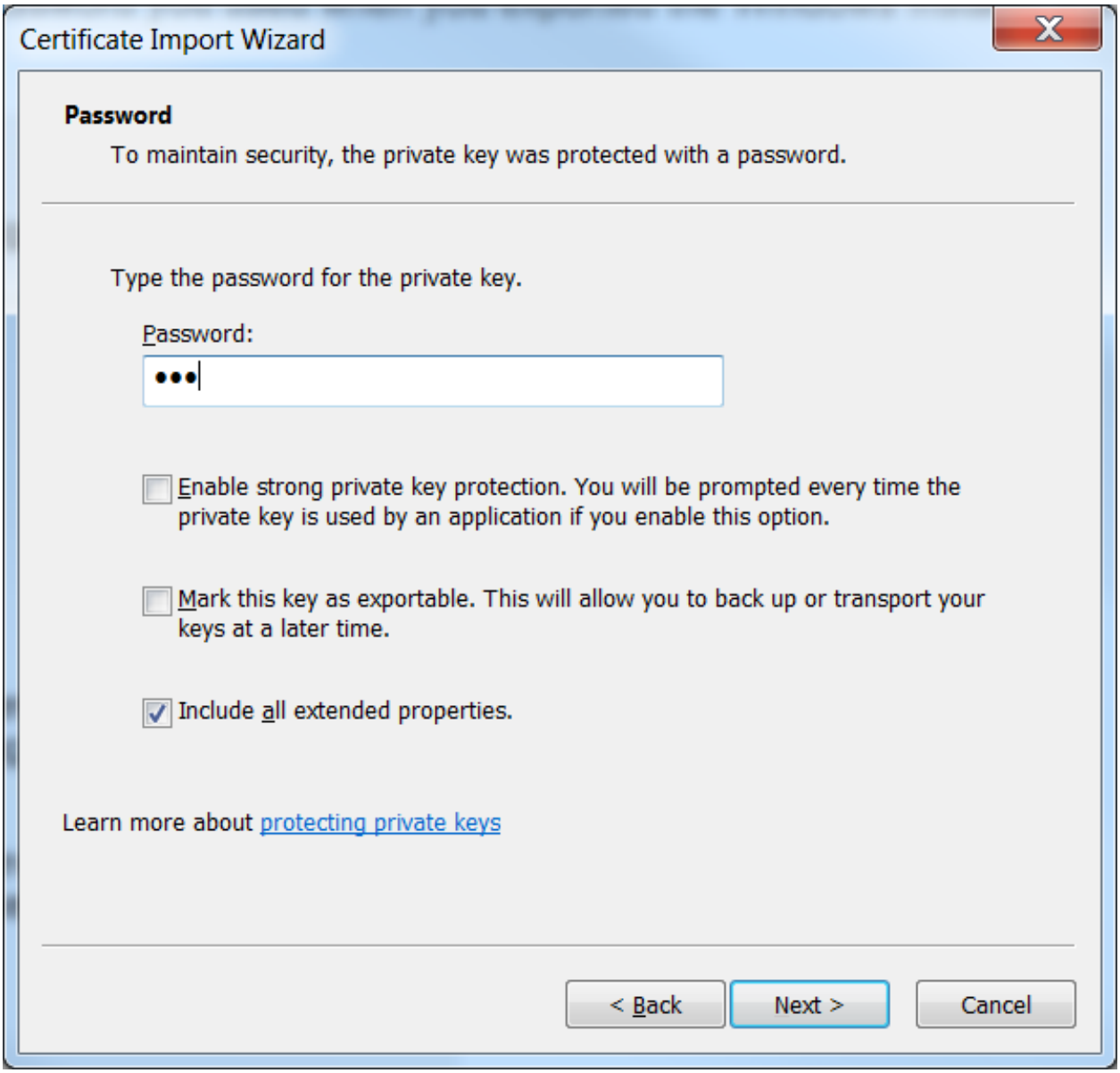
Run the installer with all defaults. When selecting components, make sure they are all checked (they are by default).



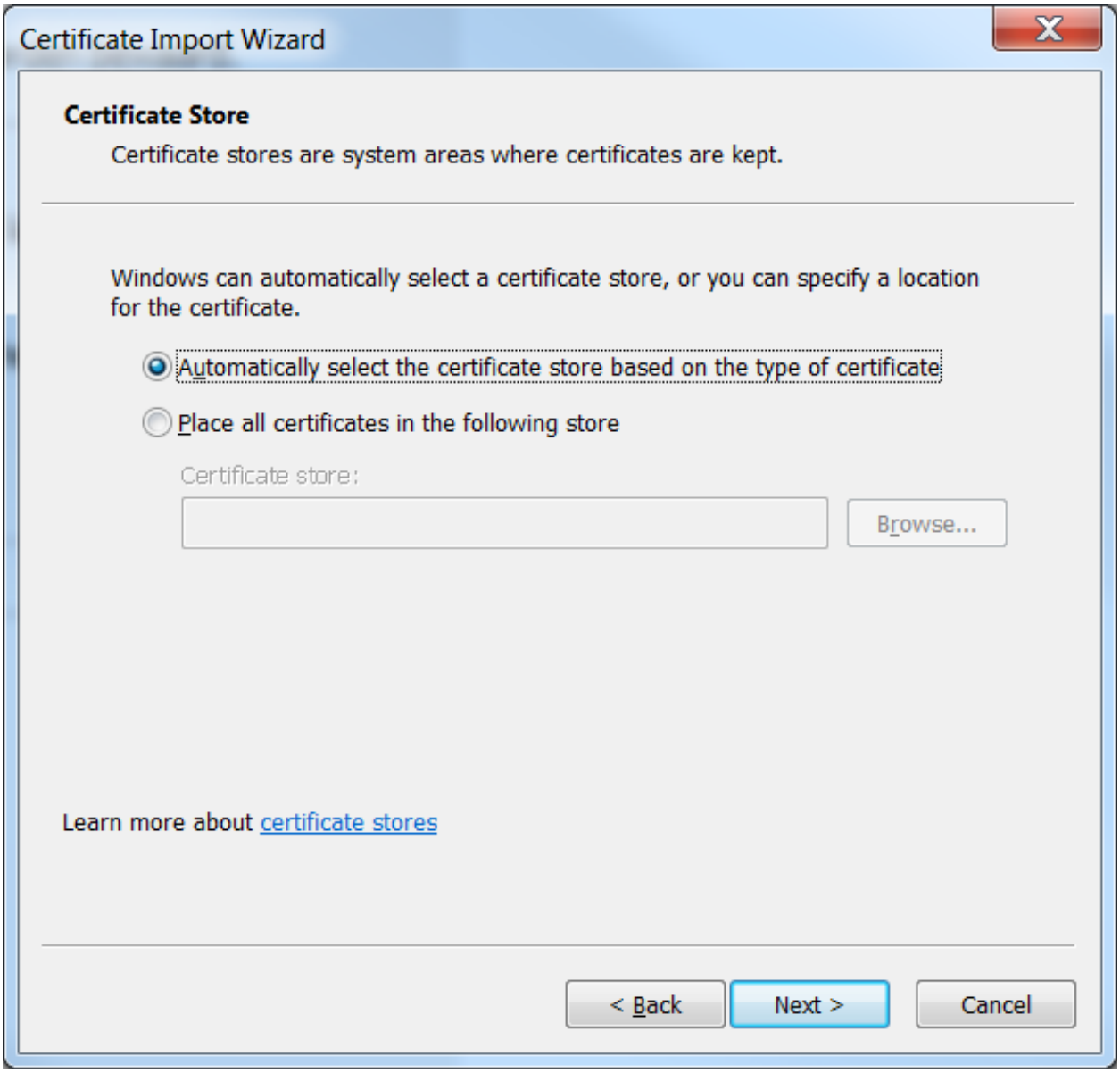
The OpenVPN Configuration Setup will continue to install the certificates.



Stick to the defaults. When prompted for a password, enter the password used when exporting the Windows Installer from the **Client Export** tab.



Have the wizard automatically select the archive.



34.51.15 Change the cryptoapicert SUBJ

Open `C:\Program Files\OpenVPN\config\config.ovpn` or `C:\Program Files(x86)\OpenVPN\config\config.ovpn` and change the line that says

cryptoapicert "SUBJ:"

to

cryptoapicert "SUBJ:username"

. . . replace `username` by the user's actual username.

This helps specifying which certificate OpenVPN should use in case certificates have a naming conflict.

34.51.16 Using the Windows client

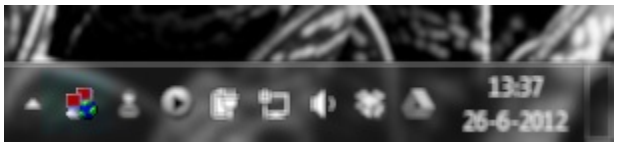
Set the Windows Client to *run as Administrator*.

To use the client, double click the OpenVPN GUI icon on the Desktop.

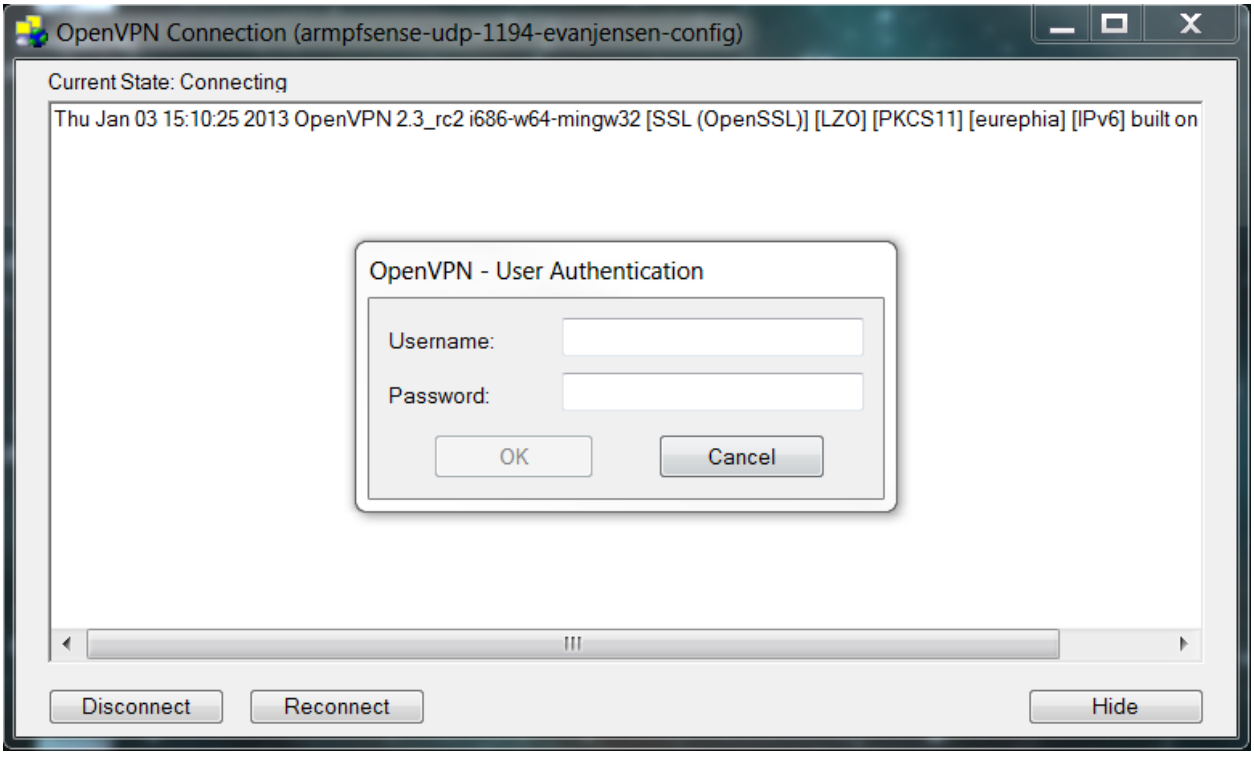


Windows will ask to confirm the execution. Confirm.

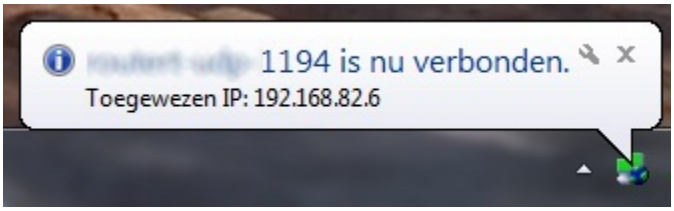
OpenVPN will start but that's not enough. Right-click the OpenVPN icon in the taskbar and choose **Connect**.



The user must now enter their username and password. This is only the username part, without the domain. The password is the user's Active Directory password.



If all is well, OpenVPN will connect to the pfSense router and minimize to the system tray.



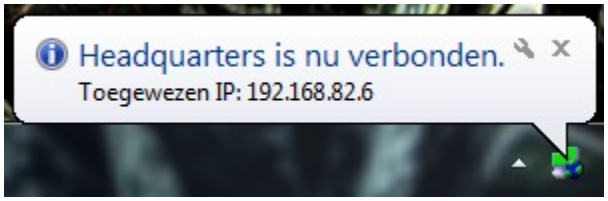
Right-click the system tray icon and choose **Disconnect** or **Close** to either disconnect the tunnel or close the OpenVPN program altogether.

Tweaking the client

Here are some tweaks I like to do on my client installations.

34.51.17 Change the name of the .ovpn file

When connecting to the firewall OpenVPN shows a balloon announcing that the VPN is up. It contains a rather cryptic Windows Installer name, but that can be changed to something more appropriate by renaming the `.ovpn` file in `C:\Windows\Program Files\OpenVPN\config` (or `C:\Windows\Program Files(x86)\OpenVPN\config`) to whatever name the balloon should show.



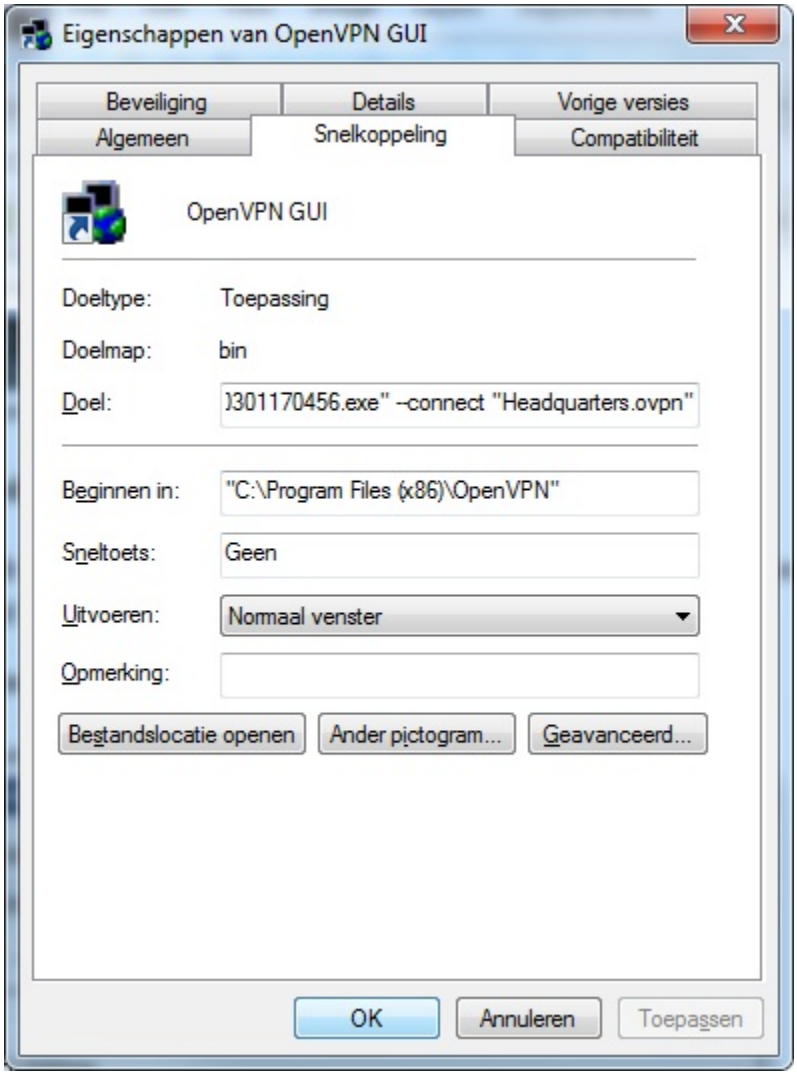
(*is nu verbonden* is dutch for *is now connected*.)

34.51.18 Edit the shortcut to connect directly

The shortcut to OpenVPN GUI can be edited to directly connect to a firewall instead of first starting OpenVPN and then starting the connection by right-clicking the shortcut and adding to the **Target** field:

> –connect "Headquarters.ovpn"

. . . if *Headquarters.ovpn* is the name of the *.ovpn* file.



The user will still need to enter their password but it does save a step in the process.

34.51.19 Edit more settings

More information on automation, customization and registry tweaks are available in this text document: http://openvpn.se/install.txt

Troubleshooting

If something doesn't work, here are some pointers for troubleshooting:

- The username may be case sensitive.
- Use the fine pfSense logging system under **Status > System logs > OpenVPN**.
- Ask questions in the pfSense forum.
- Windows 7 sometimes adds a *Microsoft Virtual WiFi Miniport Adapter*. Disabling this sometimes solves vague connection problems where there should be none.
- Is the subnet unique? Perhaps the user is in a subnet that is the same as the virtual or corporate subnet.

- Certificate problems? Check **certmgr.msc**. Perhaps an old certificate is blocking the installation of a new certificate.
- Client getting disconnected? Check the user's wifi connection. No wifi=no internet=no vpn.
- Check if the domain controller allows UDP ports 1812 and 1813 throughout the firewall. Adding the *Network Policy and Access Services* role and configuring a RADIUS client should automatically have entered these rules in the server's firewall. They are called *Network Policy Server (RADIUS Accounting - UDP-In)* and *Network Policy Server (RADIUS Authentication - UDP-In)*. Note that this is about the firewall on the domain controller, not the firewall on pfSense!

34.52 Installing OpenVPN Remote Access Clients

An OpenVPN client needs to be installed on most end-user devices, as the client functionality is not yet built into most operating systems. This section provides an overview of installation on several common operating systems.

OpenVPN Client Export Package is used to automatically generate client configuration files and installation bundles.

See also:

pfSense Hangouts on Youtube to watch the September and October 2015 Hangouts on Remote Access VPNs which covers client installations for most operating systems.

34.52.1 Installing the OpenVPN Client on Windows

The OpenVPN project provides an installer for Windows 2000 through Windows 10, downloadable from The OpenVPN Community Downloads Page. Alternately, use *OpenVPN Client Export Package* to create an installer bundled with an appropriate configuration file.

The installation is straightforward, accept all the defaults. The installation will create a new **Local Area Connection** on the client system for the OpenVPN TAP adapter. This interface will appear connected when the VPN is established and will otherwise show as disconnected. No configuration of this interface is necessary as its configuration will be pulled from the OpenVPN server or client configuration.

Current stable versions of the OpenVPN client for Windows utilize a system service so the GUI can run without elevated privileges. Once installed, a user does not need administrative access to run the client.

34.52.2 Installing the OpenVPN Client on Mac OS X

There are three client options for Mac OS X.:

- The OpenVPN command line client. Most users prefer a graphical client, so this option will not be covered.
- Tunnelblick, a free option available for download at the Tunnelblick Website.
- The commercial Viscosity client. At the time of this writing, it costs $14 USD for a single seat. If OpenVPN is used frequently, Viscosity is a much nicer client and well worth the cost.

Both Tunnelblick and Viscosity are easily installed, with no configuration options during installation.

Configuring Viscosity

When using the Viscosity client, it can be configured manually or the OpenVPN Client Export package may be used to import the configuration. Viscosity provides a GUI configuration tool that can be used to generate the underlying OpenVPN client configuration. The CA and certificates can be imported manually, and all of the parameters can be set by hand. This section cover importing a Viscosity bundle from the export package.

- Download a copy of the **Viscosity bundle** for the client from the OpenVPN Client Export package
- Locate the saved file, which will end in `.visc.zip` indicating that it is a compressed archive
- Copy this exported bundle to a folder on the Mac
- Double click this file and it will expand to `Viscosity.visc`
- Double click `Viscosity.visc` and Viscosity will open and import the connection as shown in Figure *Viscosity Import*

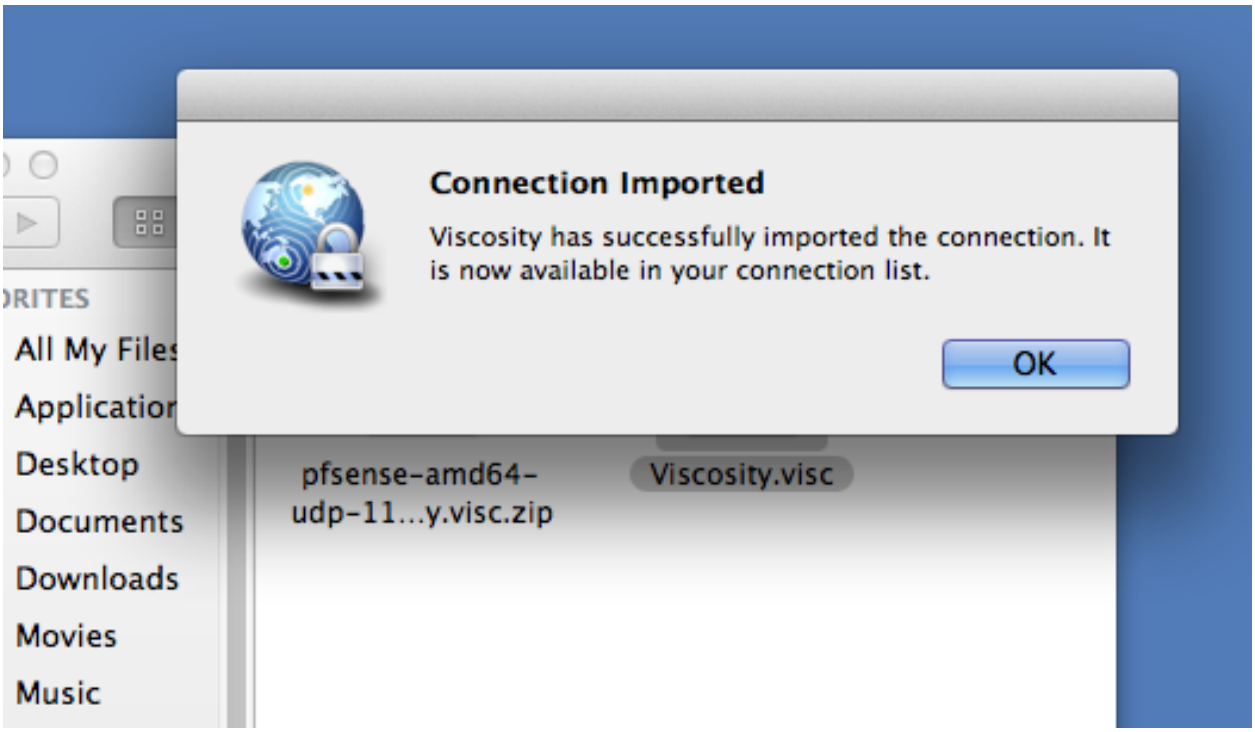


Fig. 71: Viscosity Import

- Delete the `Viscosity.visc` directory and the `.zip` archive
- Viscosity will be running after import, and may be found in the menu bar
- Click the lock icon added to the menu bar at the top of the screen
- Click **Preferences** to check that the configuration was imported as shown in Figure *Viscosity Preferences*
- Check the **Connections** area to see if the connection imported successfully as shown in Figure *Viscosity View Connections*.
- Close the Preferences screen
- Click the lock in the menu bar

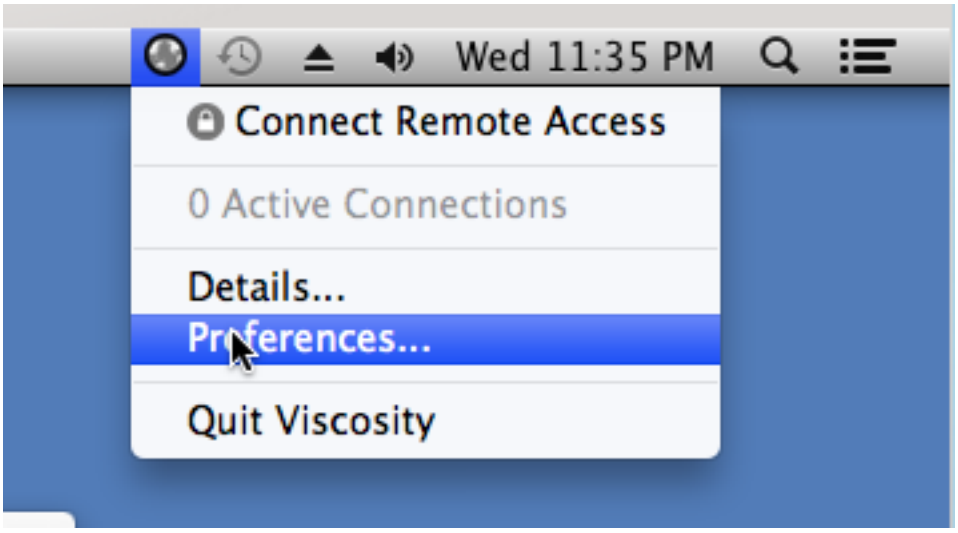


Fig. 72: Viscosity Preferences

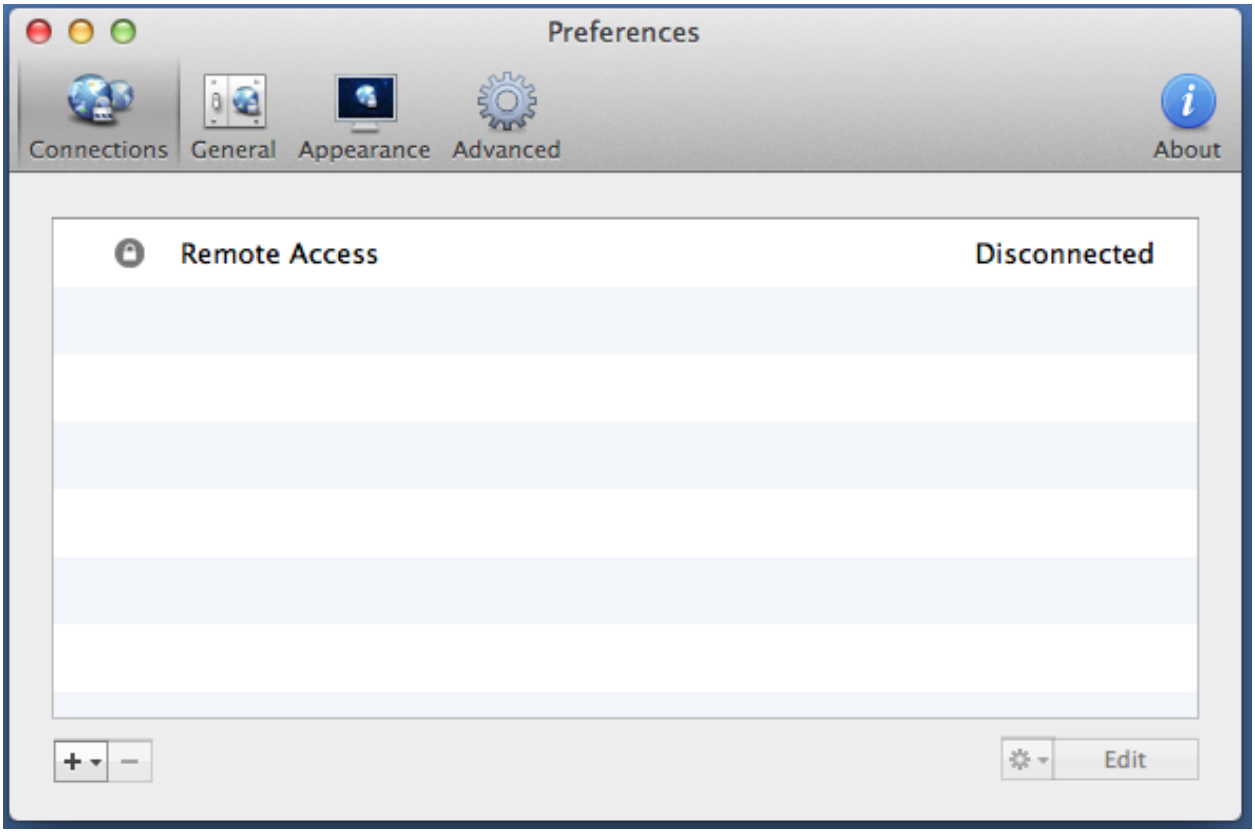


Fig. 73: Viscosity View Connections

- Click the name of the VPN connection to connect as shown in Figure *Viscosity Connect*. After a few seconds, the lock in the menu bar will turn green to show it connected successfully.

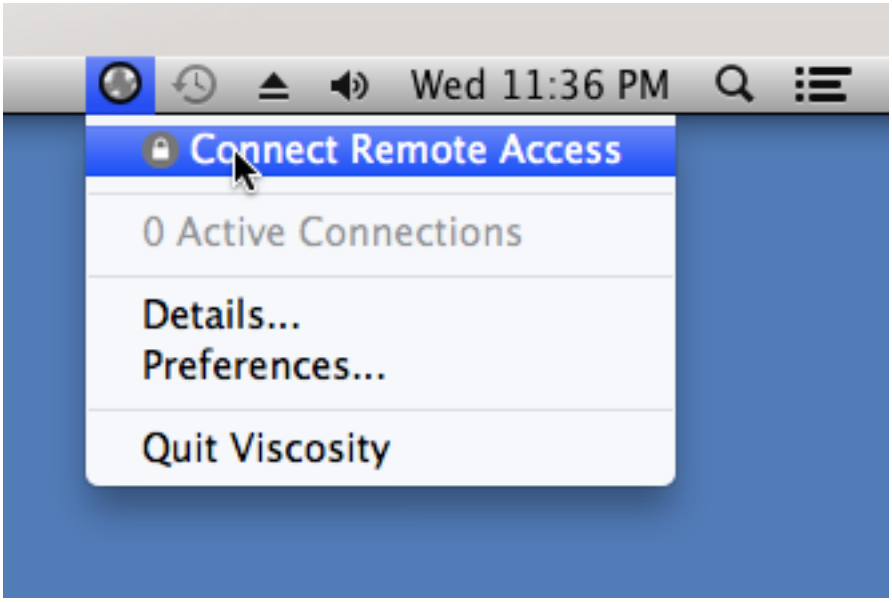


Fig. 74: Viscosity Connect

- Click on it and then click **Details** as shown in Figure *Viscosity Menu* to see connection information

On the first screen (Figure *Viscosity Details*), the connection status, connected time, the IP assigned to the client, and the IP of the server are all displayed. A bandwidth graph is displayed at the bottom of the screen, showing the throughput in and out of the OpenVPN interface.

Clicking the up/down arrow button in the middle of the details screen displays additional network traffic statistics. This shows the traffic sent within the tunnel (TUN/TAP In and Out), as well as the total TCP or UDP traffic sent including the overhead of the tunnel and encryption. For connections using primarily small packets the overhead is considerable with all VPN solutions. The stats shown in Figure *Viscosity Details: Traffic Statistics* are from only a few pings traversing the connection. The traffic sent in bringing up the connection is also counted here, so the initial overhead is higher than what it will be after being connected for some time. Also, the typical VPN traffic will have larger packet sizes than 64 byte pings, making the total overhead and difference between these two numbers considerably less.

Clicking on the third icon in the middle of the **Details** screen shows the OpenVPN log file (Figure *Viscosity Details: Logs*). If there is any trouble connecting, review the logs here to help determine the problem. See also *Troubleshooting OpenVPN*.

34.52.3 Installing the OpenVPN Client on iOS

iOS is also capable of running OpenVPN natively using the iOS OpenVPN Connect client available in the App Store. This app does not require jailbreaking the iOS device. The app must have the config file and certificates configured outside of the iOS device and then imported to it. The OpenVPN Client Export package on pfSense® can be used to export an **OpenVPN Connect** type **Inline Configuration**. Transfer the resulting `.ovpn` file to the target device then by using iTunes to transfer the files into the app or e-mail it to the device.

Using other methods to get files onto the device remotely, such as Dropbox, Google Drive, or Box will work similarly to the e-mail method are generally more secure as the contents will remain private and possibly encrypted depending on the method and storage.

If using the e-mail method, use the following procedure:

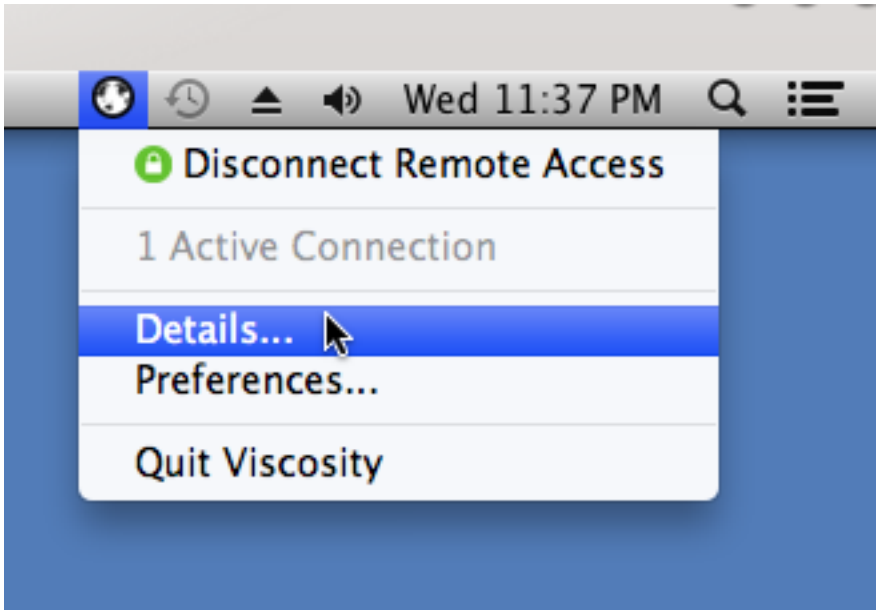


Fig. 75: Viscosity Menu

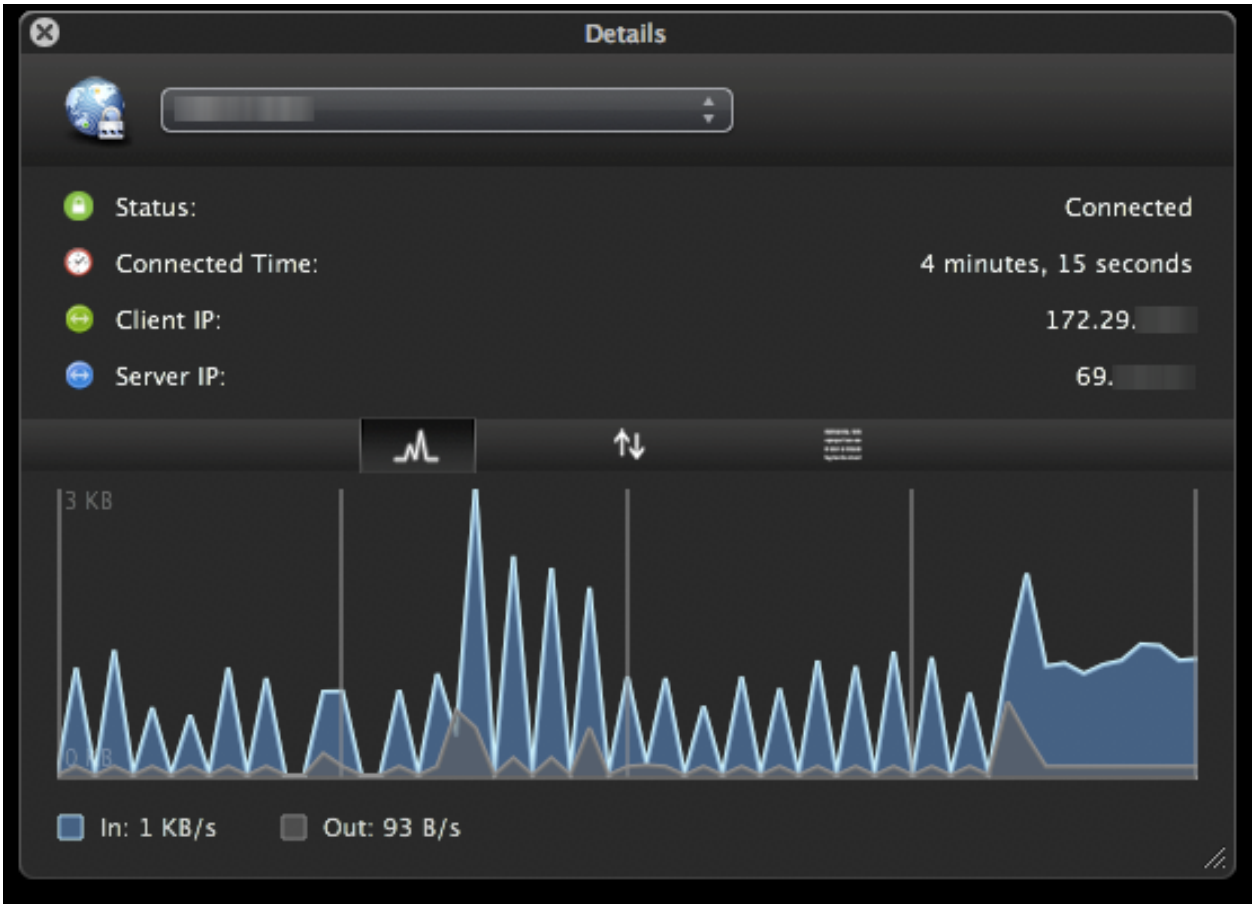


Fig. 76: Viscosity Details

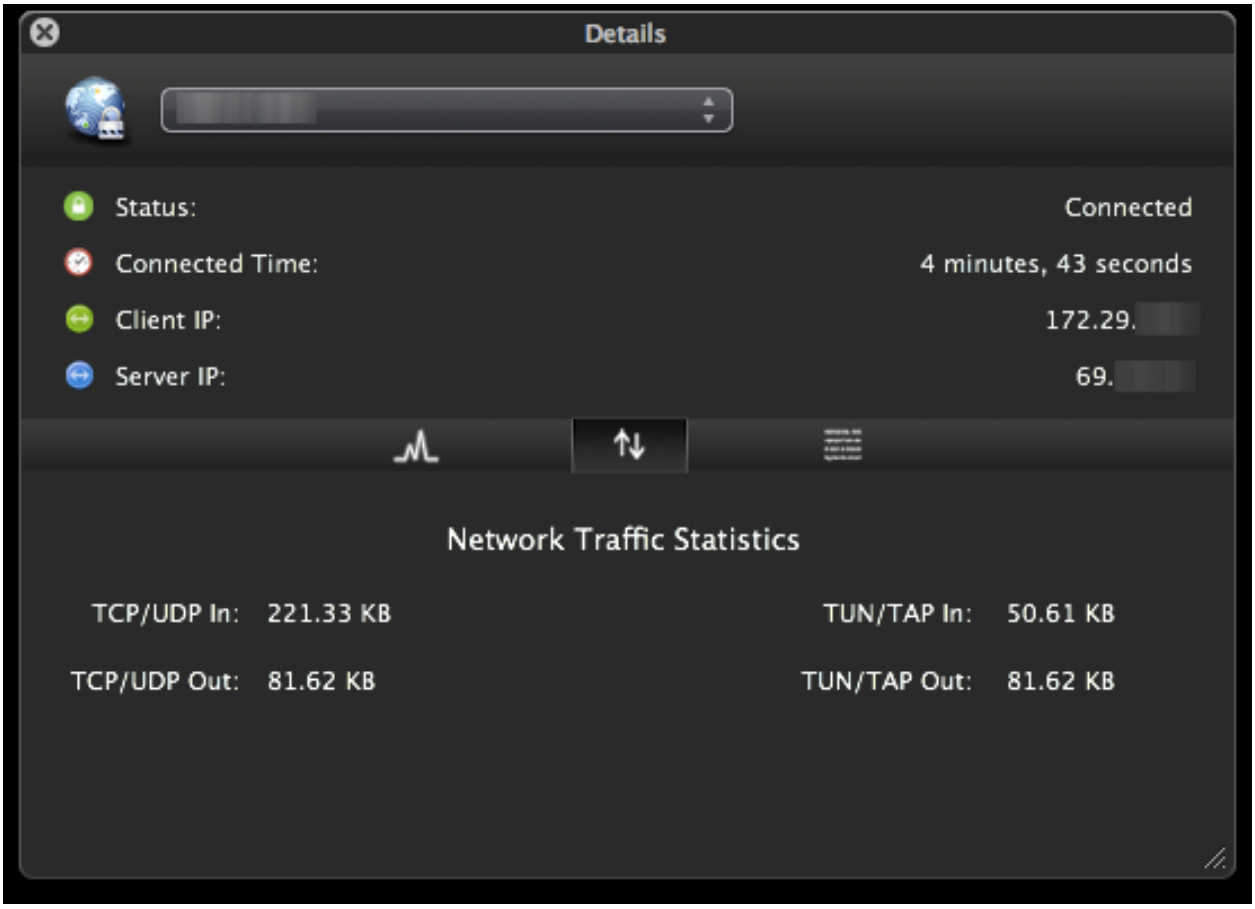


Fig. 77: Viscosity Details: Traffic Statistics

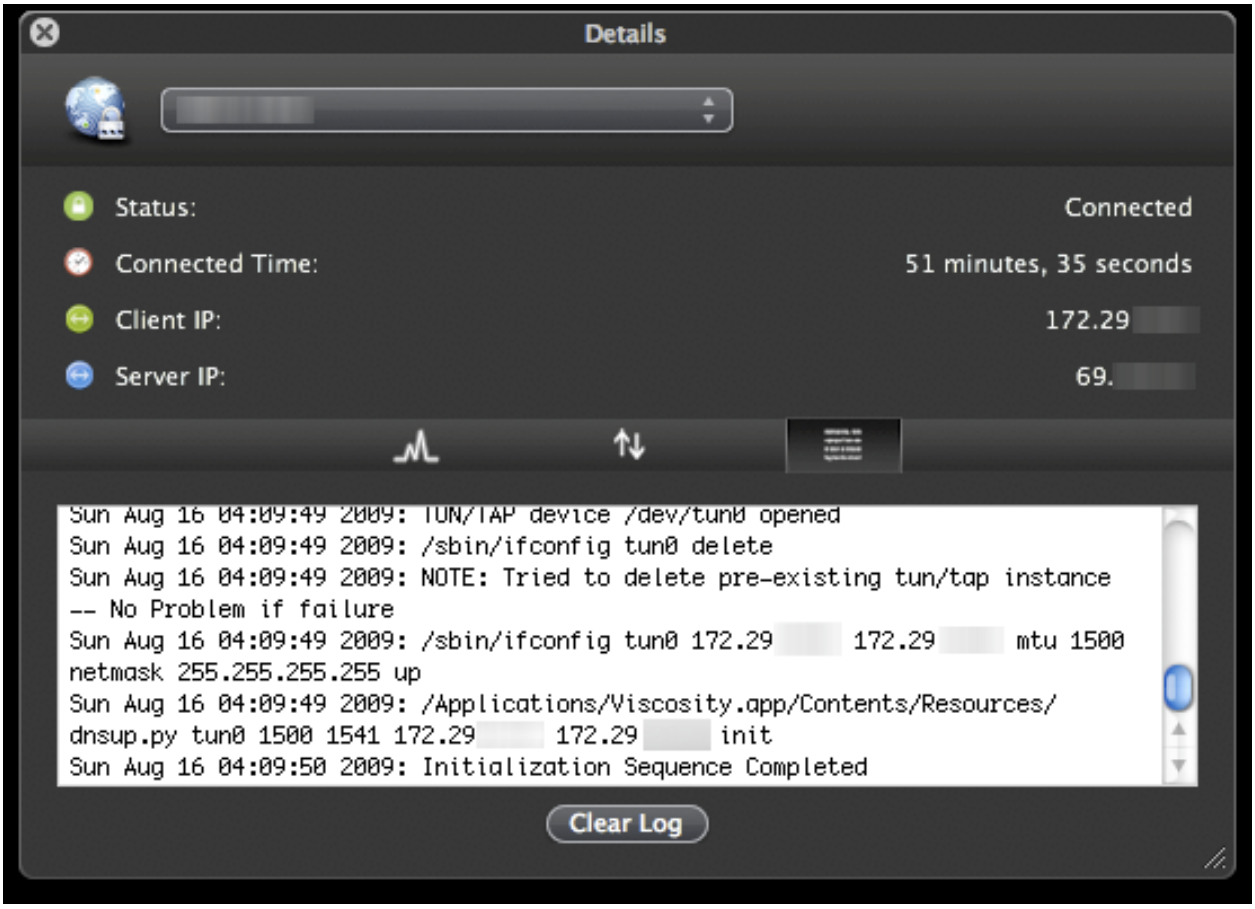


Fig. 78: Viscosity Details: Logs

- Export the **OpenVPN Connect** type **Inline Configuration** file for the VPN.
- Send the exported file in an e-mail to an account configured on the iOS device
- Install the OpenVPN Connect app on the device
- Open the Mail app on the device
- Open the e-mail message containing the attachment
- Tap the attachment. When it is tapped one of the choices will be to open it with the OpenVPN Connect app
- Tap to select the OpenVPN connect app and it will offer to import the configuration
- Tap the + button and the profile will be imported

Using iTunes to transfer the configuration to the iOS device is simple and more secure than e-mail.

- Export the **OpenVPN Connect** type **Inline Configuration** file for the VPN.
- Connect the iOS device to the computer and open iTunes
- Find and install the OpenVPN Connect app
- Click the device icon inside of iTunes in the toolbar
- Select **Apps** on the left side of the window
- Locate the **File Sharing** section At the bottom of this screen (scroll down)
- Click the icon for OpenVPN under **File Sharing** and a list of files will show on the right under the heading **OpenVPN Documents**
- Copy the file to the device by using ONE of the following methods. The file will be immediately available on the iOS device.
  - Use Finder to drag and drop the *.ovpn* file into this area -OR-
  - Click **Add** and locate the file to import
- Open the OpenVPN Connect app and it will offer to import the profile
- Tap the + button, and the profile will be imported

If the profile is configured for user authentication it will prompt for the credentials, which may optionally be saved. Underneath the credential prompt is a connection status which will change between **Disconnected** and **Connected** and also indicates when a connection is being attempted. Clicking this will open the OpenVPN client log which is very useful if connection problems are encountered.

To connect the VPN, move the slider at the bottom of the profile from **Off** to **On** and the app will attempt to connect. To manually disconnect, move the slider back to **Off**.

When manually building a configuration file for this client it requires either an inline configuration style or separate CA, client certificate, client certificate key, and TLS key files (if used). It does not appear to accept *.p12* files containing the CA and client certificate/keys, so the default "Configuration Archive" style will not work, though some users have reported success importing the configuration files extracted from the Viscosity bundle.

34.52.4 Installing the OpenVPN Client on Android

For devices running Android 4.0 or a newer release, there is a free OpenVPN app in the Google Play store that works excellently without needing root access. It is called OpenVPN for Android by Arne Schwabe.

OpenVPN Client Export Package on pfSense® can export an **Android** type **Inline Configuration**, and the resulting `.ovpn` file can be transferred to the target device. It can be copied directly, e-mailed to the device, etc.

- Open the **OpenVPN for Android** app
- Tap Import (File folder icon at upper right)
- Find the `.ovpn` file saved above and tap it
- Tap Import (Disk icon at upper right)

The imported VPN is now shown in the list. Edit the entry to change the name and other details. Tap the VPN to connect. If the profile is configured for user authentication, the app will prompt for credentials when connecting.

Note: The Android OpenVPN Connect client also works on Android 4.x and does not require root. It works identically to the iOS client by the same name. It lacks the ability to fully configure the VPN in the GUI, so it is not recommended. Use the **OpenVPN Connect** type **Inline Configuration** export for use with that client on both Android and iOS.

34.52.5 Installing the OpenVPN Client on FreeBSD

If the client has a stock FreeBSD installation, OpenVPN may be found in the ports collection.

To install OpenVPN, run the following as **root**:

```
# pkg install openvpn
```

Alternately, it can be installed from ports:

```
# cd /usr/ports/security/openvpn && make install clean
```

34.52.6 Installing the OpenVPN Client on Linux

Installing OpenVPN on Linux will vary depending on the preferred distribution and method of managing software installations. OpenVPN is included in the package repositories of most major Linux distributions. With all the various possibilities between countless distributions, and adequate information already available in other sources online, this documentation will not cover specifics. Simply search the Internet for the distribution of choice and "installing OpenVPN" to find information.

Ubuntu-based distributions have OpenVPN management integrated with Network Manager, but it requires installing an extra module.

34.52.7 Installing the OpenVPN Client Configuration Manually

Performing a manual client installation instead of using the *OpenVPN Client Export Package* requires additional steps to install the software and settings onto the client devices. Installing the client on other operating systems is left up to the reader.

After installing OpenVPN, copy the certificates to the client and create the client configuration file.

Copy certificates

Three files from the firewall are needed for each client: the CA certificate, the client certificate, and the client key. A fourth file, the TLS key, is only required if the server was configured for TLS authentication.

- Export the CA certificate from **System Cert > Manager** on the **CAs** tab, save this as `ca.crt`
- Export the client certificate and key as described in *Local Users*, save these as `username.crt` and `username.key`
- Copy these files to the OpenVPN `config` directory on the client
- Copy the TLS key from the server configuration screen If TLS authentication is used on this OpenVPN server. Save this into a new text file called `tls.key` and include it in the `config` folder as well.

Create Configuration

After copying the certificates to the client, the OpenVPN client configuration file must be created. This can be done with any plain text file editor such as Notepad on Windows. The following shows the options most frequently used:

```
client
dev tun
proto udp
remote vpn.example.com 1194
ping 10
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert username.crt
key username.key
verb 3
comp-lzo
tls-auth tls.key 1
auth-user-pass
```

remote Specifies the host and port of the remote OpenVPN server. An IP address or FQDN can be specified here.

proto Specifies the protocol used by the OpenVPN connection. Change this line to `proto tcp` if TCP is used on the OpenVPN server.

ca, cert, key Must be modified accordingly for each client to reflect the filenames saved previously.

tls-auth If TLS authentication is not used, the tls-auth line may be omitted.

auth-user-pass If the remote access VPN does not include username and password authentication, omit this line.

See also:

For a more complete reference on the OpenVPN directives, refer to the OpenVPN manual for the installed client version.

Distributing configuration and keys to clients

The easiest way to distribute the keys and OpenVPN configuration to clients is via the OpenVPN Client Export package. If that package is not a viable choice, place the needed files in a ZIP archive or self-extracting archive automatically extracting to `C:\Program Files\OpenVPN\config`. This must be transmitted securely to the end user, and must never be passed over untrusted networks unencrypted.

34.53 Adding OpenVPN Remote Access Users

At this point the VPN server is configured but there may not be any clients which can connect. The method for adding users to the VPN will depend upon the authentication method chosen when creating the OpenVPN server.

See also:

More details on adding users can be found in *User Management and Authentication*. More information on managing user certificates can be found in *User Certificates*.

34.53.1 Local Users

To add a user that can connect to OpenVPN, they must be added to the User Manager as follows:

- Navigate to **System > User Manager**
- Click  **Add** to create a new user
- Enter a **Username**, **Password**, and password confirmation
- Fill in **Full Name** (optional)
- Check **Click to create a user certificate**, which will open the certificate options panel
- Enter the user's name or some other pertinent information into the **Descriptive Name** field
- Choose the same **Certificate Authority** used on the OpenVPN server
- Choose a **Key Length** (may be left at the default)
- Enter a **Lifetime** (may be left at the default)
- Click **Save**

To view or change the user:

- Navigate to **System > User Manager**
- Click  next to the row containing the user to see/edit

To export a user's certificate and key:

Note: This part may be skipped if using the OpenVPN Client Export Package, described in *OpenVPN Client Export Package*. The client export package is a much easier way to download client configurations and installation files.

- Navigate to **System > Cert Manager** on the **Certificates** tab
- Locate the user certificate in the list
- Click  to download the user certificates
- Click  to download the key for the certificate
- Click  to download a PKCS#12 bundle which includes the user certificate and key, and the CA Certificate (optional).

In most cases, the CA Certificate should also be downloaded with the user certificate. This can be done from its entry on **System > Cert Manager**, **CAs** tab, or by using the PKCS#12 bundle mentioned previously.

34.53.2 LDAP or RADIUS Users

Adding LDAP and RADIUS users will fully depend on the server implementation and management tools, which are beyond the scope of this documentation. Contact the server administrator or software vendor for assistance. Certificates for LDAP or RADIUS users cannot be created from within the firewall's web interface in a way that reflects a user-certificate relationship. However, it is possible to create the certificates on their own using the certificate manager as described in *User Certificates*

34.54 OpenVPN Site-to-Site Configuration Example with Shared Key

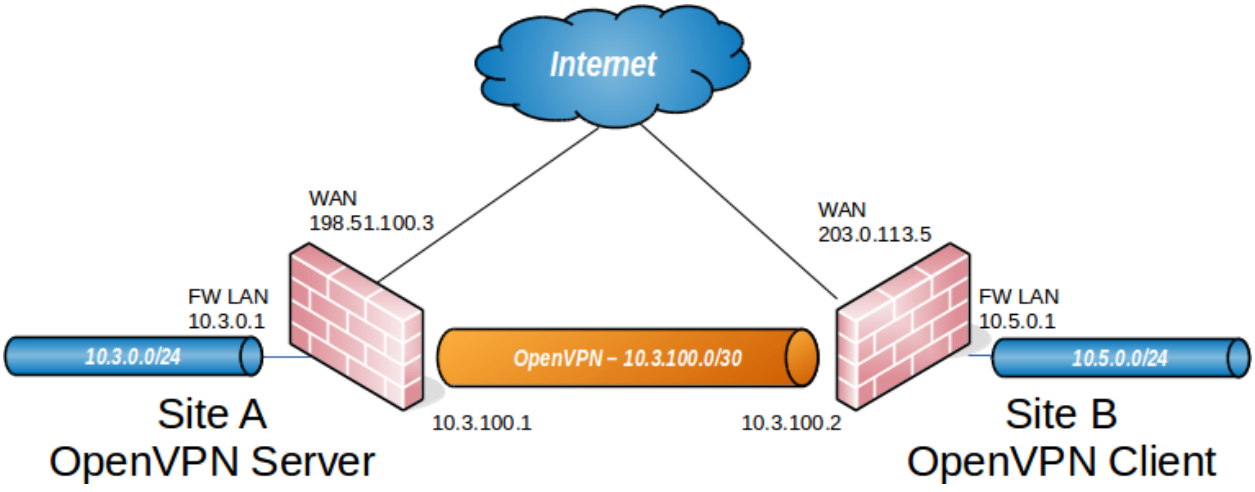


Fig. 79: OpenVPN Example Site-to-Site Network

This section describes the process of configuring a site-to-site connection using a **shared key** style OpenVPN tunnel.

When configuring a shared key site-to-site OpenVPN connection one firewall will be the server and the other will be the client. Usually the main location will be the server side and the remote offices will act as clients, though the opposite is

functionally equivalent. Similar to a remote access OpenVPN configuration there will be a dedicated subnet in use for the OpenVPN interconnection between networks in addition to the subnets on both ends. The example configuration described here is depicted in Figure *OpenVPN Example Site-to-Site Network*.

`10.3.100.0/30` is used as the **Tunnel Network**. The OpenVPN tunnel between the two firewalls gets an IP address on each end out of that subnet, as illustrated in the diagram. The following sections describe how to configure the server and client sides of the connection.

34.54.1 Configuring Server Side

- Navigate to **VPN > OpenVPN**, **Server** tab
- Click  **Add** to create a new server entry
- Fill in the fields as follows, with everything else left at defaults:

  Server Mode Select *Peer to Peer (Shared Key)*.

  Description Enter text here to describe the connection (e.g. `ExampleCo Site B VPN`)

  Shared key Check **Automatically generate a shared key**, or paste in a pre-existing shared key for this connection.

  Tunnel Network Enter the previously chosen network, `10.3.100.0/30`

  Remote network Enter the LAN on the Site B side, `10.5.0.0/24`
- Click **Save**
- Click  to edit the server that was created a moment ago
- Find the **Shared Key** box
- Select all text inside the **Shared Key** box
- Copy the text to the clipboard
- Save the contents to a file, or paste into a text editor such as Notepad temporarily

Next, add a firewall rule on WAN allowing access to the OpenVPN server.

- Navigate to **Firewall > Rules**, **WAN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *UDP*
- Set the Source address to match the client. If it has a dynamic IP address, leave it set to *Any*, otherwise set the rule to only allow from the WAN IP address of the client:
  - Select *Single Host or Alias* in **Source**
  - Enter the WAN address of the client as the **Source address** (e.g. `203.0.113.5`)
- Set the **Destination** to *WAN Address*
- Set the **Destination port** to `1194` in this instance
- Enter a **Description**, such as `OpenVPN from Site B`
- Click **Save** and the rule will look like Figure *OpenVPN Example Site-to-Site WAN Firewall Rule*.

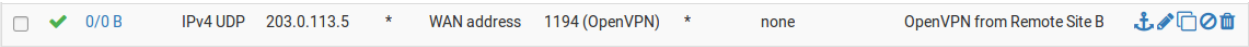


Fig. 80: OpenVPN Example Site-to-Site WAN Firewall Rule

- Click **Apply Changes**

A rule must also be added to the **OpenVPN** interface to pass traffic over the VPN from the Client-side LAN to the Server-side LAN. An "Allow all" style rule may be used, or a set of stricter rules. In this example allowing all traffic is OK so the following rule is made:

- Navigate to **Firewall > Rules**, **OpenVPN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *any*
- Enter a **Description** such as `Allow all on OpenVPN`
- Click **Save**
- Click **Apply Changes**

The server configuration is finished.

34.54.2 Configuring Client Side

- Navigate to **VPN > OpenVPN**, **Client** tab on the client system
- Click  **Add** to create a new OpenVPN client instance
- Fill in the fields as follows, with everything else left at defaults:

  Server Mode Select *Peer to Peer (Shared Key)*.

  Server host or address Enter the public IP address or hostname of the OpenVPN server here (e.g. `198.51.100.3`).

  Description Enter text to describe the connection (e.g. `ExampleCo Site A VPN`)

  Shared key Uncheck **Automatically generate a shared key**, then paste in the shared key for the connection using the key copied from the server instance created previously.

  Tunnel Network Must match the server side exactly (e.g. `10.3.100.0/30`)

  Remote network Enter the LAN network on the Site A side, `10.3.0.0/24`

- Click **Save**

A rule must also be added to the **OpenVPN** interface to pass traffic over the VPN from the Server-side LAN to the Client-side LAN. An "Allow all" style rule may be used, or a set of stricter rules. In this example allowing all traffic is OK so the following rule is made:

- Navigate to **Firewall > Rules**, **OpenVPN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *any*
- Enter a **Description** such as `Allow all on OpenVPN`

- Click **Save**
- Click **Apply changes**

The configuration of the client is complete. No firewall rules are required on the client side WAN interface because the client only initiates outbound connections. The server never initiates connections to the client.

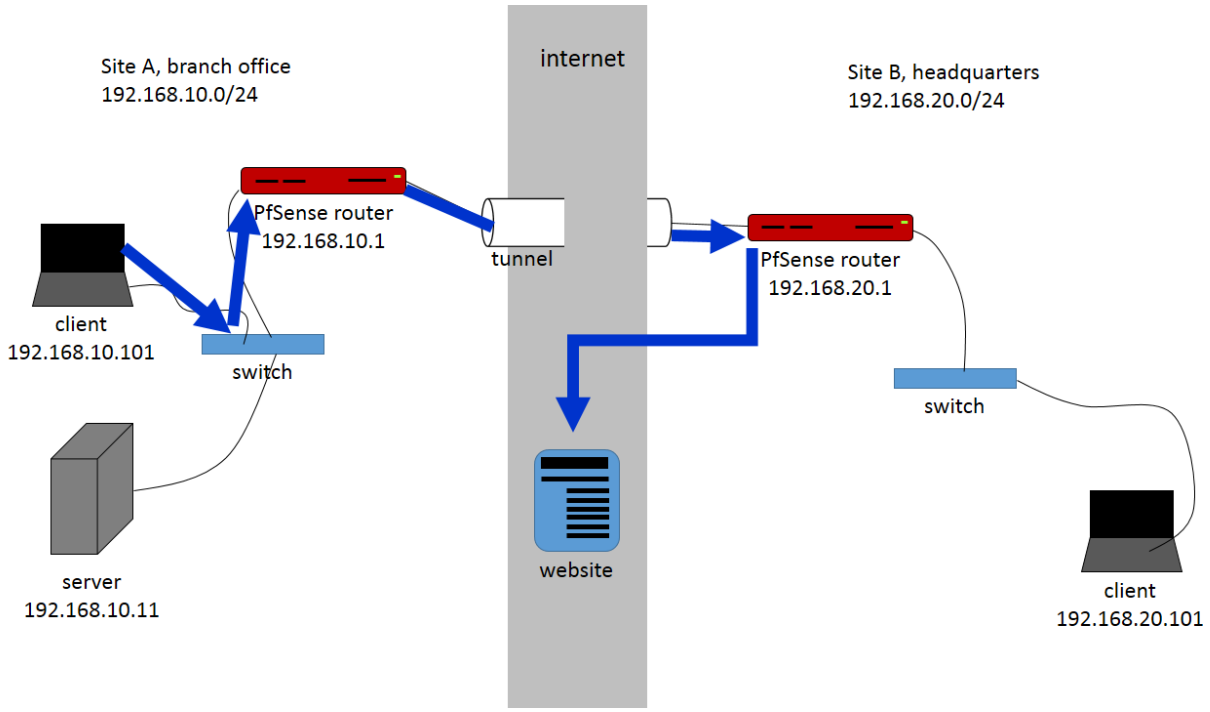
Note: With remote access PKI configurations, typically routes and other configuration options are not defined on the client configuration, but rather they are pushed from the server to the client. With shared key deployments, routes and other parameters **must** be defined on both ends as needed (as described previously, and later in *Custom configuration options*), options cannot be pushed from the server to clients when using shared keys.

34.54.3 Testing the connection

The connection will immediately be active upon saving on the client side. Try to ping across to the remote end to verify connectivity. If problems arise, refer to *Troubleshooting OpenVPN*.

34.55 Routing Internet Traffic Through A Site-To-Site OpenVPN Tunnel

This article shows how to create a site-to-site connection using OpenVPN and how to route the Internet connection of site A through site B using pfSense® software.



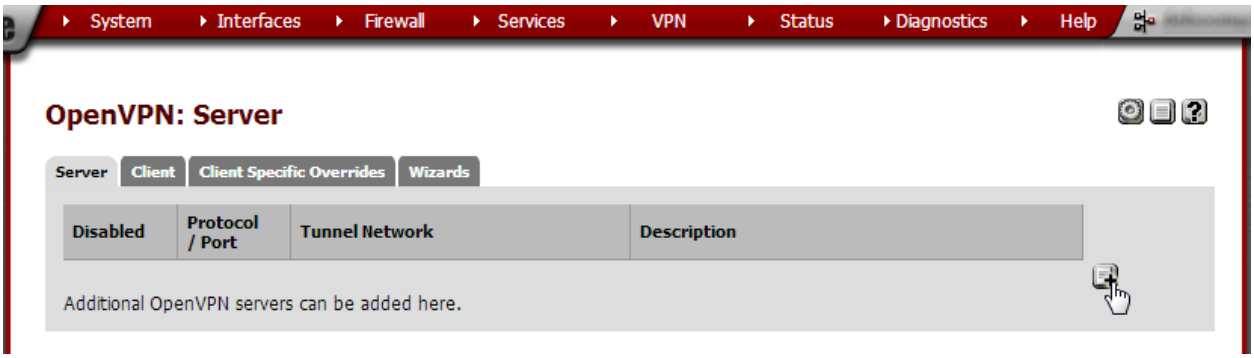
This is effectively the same as using an IPsec site-to-site connection except that we'll be using OpenVPN instead of IPsec. Using OpenVPN as the 'back-end' means we need to set up one side as a server and the other as the client. It doesn't matter which one is which but if more than two sites are connected in a star topology it seems natural to use the center of the star as the server. The server also needs to have a dedicated port mapped to it if it's behind another router, or is must reside in its DMZ.

For the purpose of this article:

- **Site A is a branch office**, LAN subnet 192.168.10.0/24
- **Site B is the main office** through which all Internet traffic is routed, 192.168.20.0/24

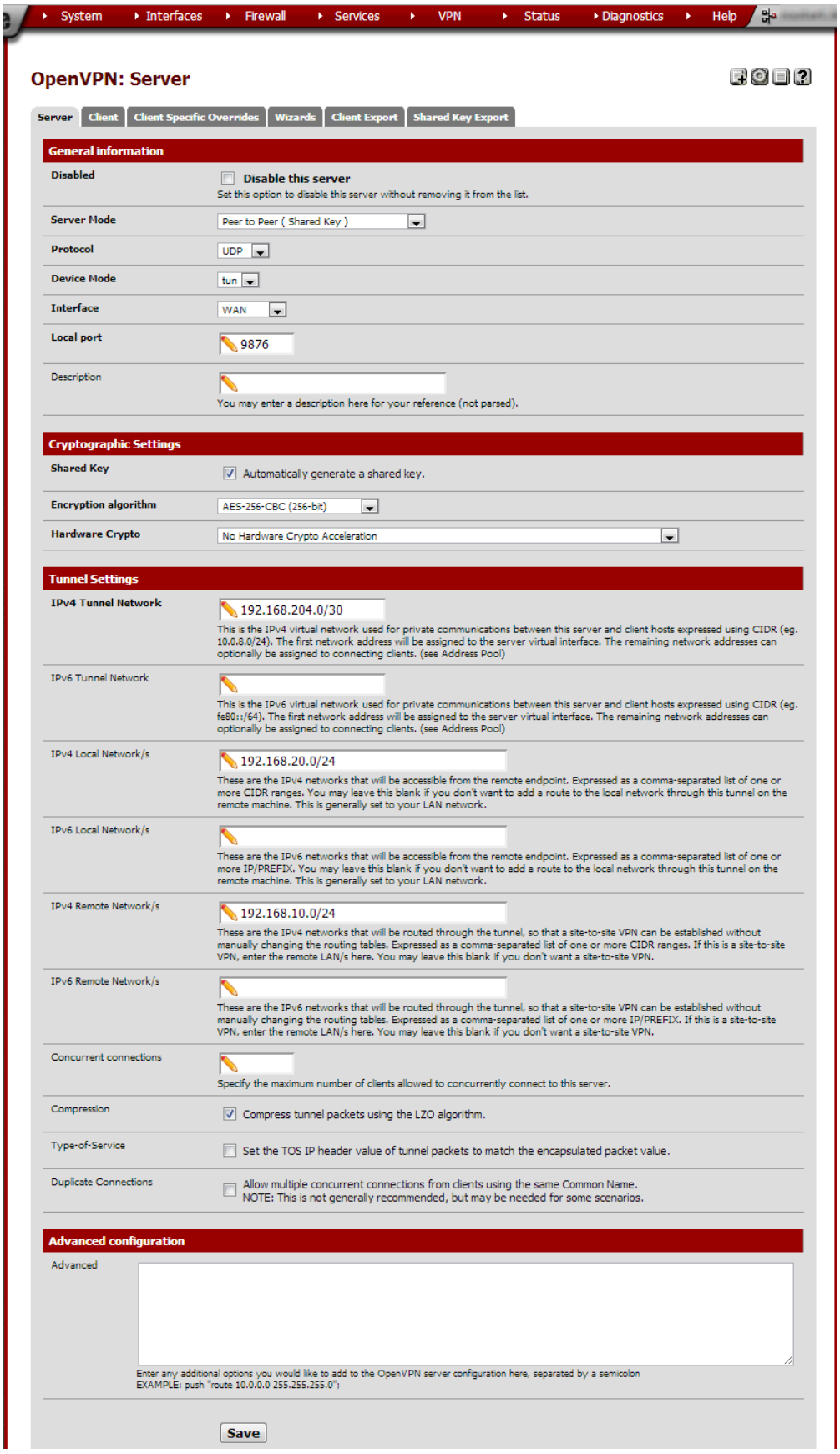
34.55.1 Set up OpenVPN at Site B

From the **VPN** menu choose **OpenVPN**. On the page under the **Server** tab, click the **+** button to create a new OpenVPN server.



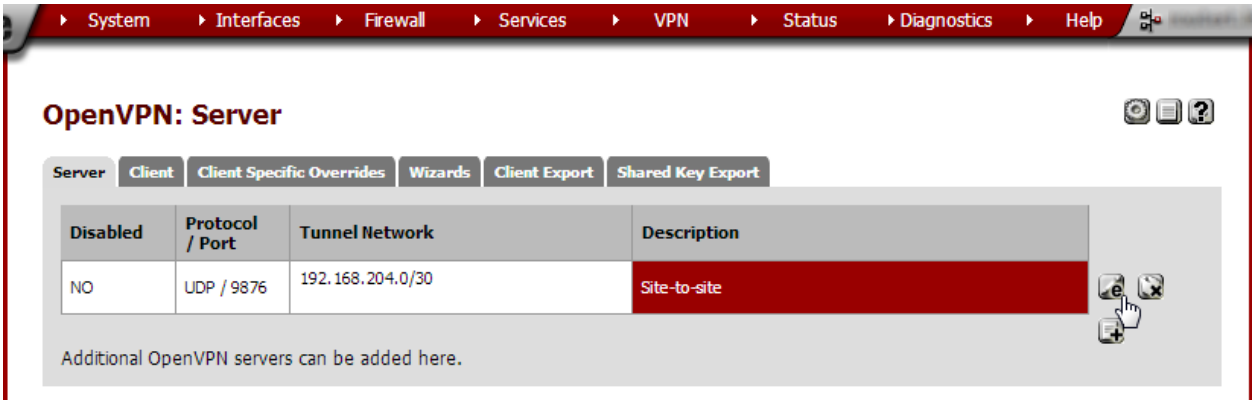
Enter these values:

Server Mode	Peer to Peer (Shared Key)	
Protocol	UDP	
Device Mode	tun	
Interface	WAN	
Local port	9876	1194 is the default OpenVPN port. It doesn't hurt to change it to another number to add some security through obscurity. Any unused port number may be used but we'll stick to 9876 in this article.
Description	Site-to-site	
Shared Key	Checked	
Encryption algorithm	AES-256-CBC (256-bit)	
Hardware Crypto	No Hardware Crypto Acceleration unless it is needed for this hardware.	If in doubt, select 'No Hardware Crypto Acceleration'.
IPv4 Tunnel Network	192.168.204.0/30	Choose a subnet that's not in use in any of the current LANs. This will be used internally by OpenVPN. We're using 192.168.204.0/30 here but any private range will do. The /30 mask is because OpenVPN will only use one IP address per site. We're connecting two sites so two addresses will suffice. /24 will work but is overkill.
IPv6 Tunnel Network	leave empty	
IPv4 Local Network/s	192.168.20.0/24	Site B's subnet
IPv6 Local Network/s	leave empty	
IPv4 Remote Network/s	192.168.10.0	Site A's subnet
IPv6 Remote	leave empty	

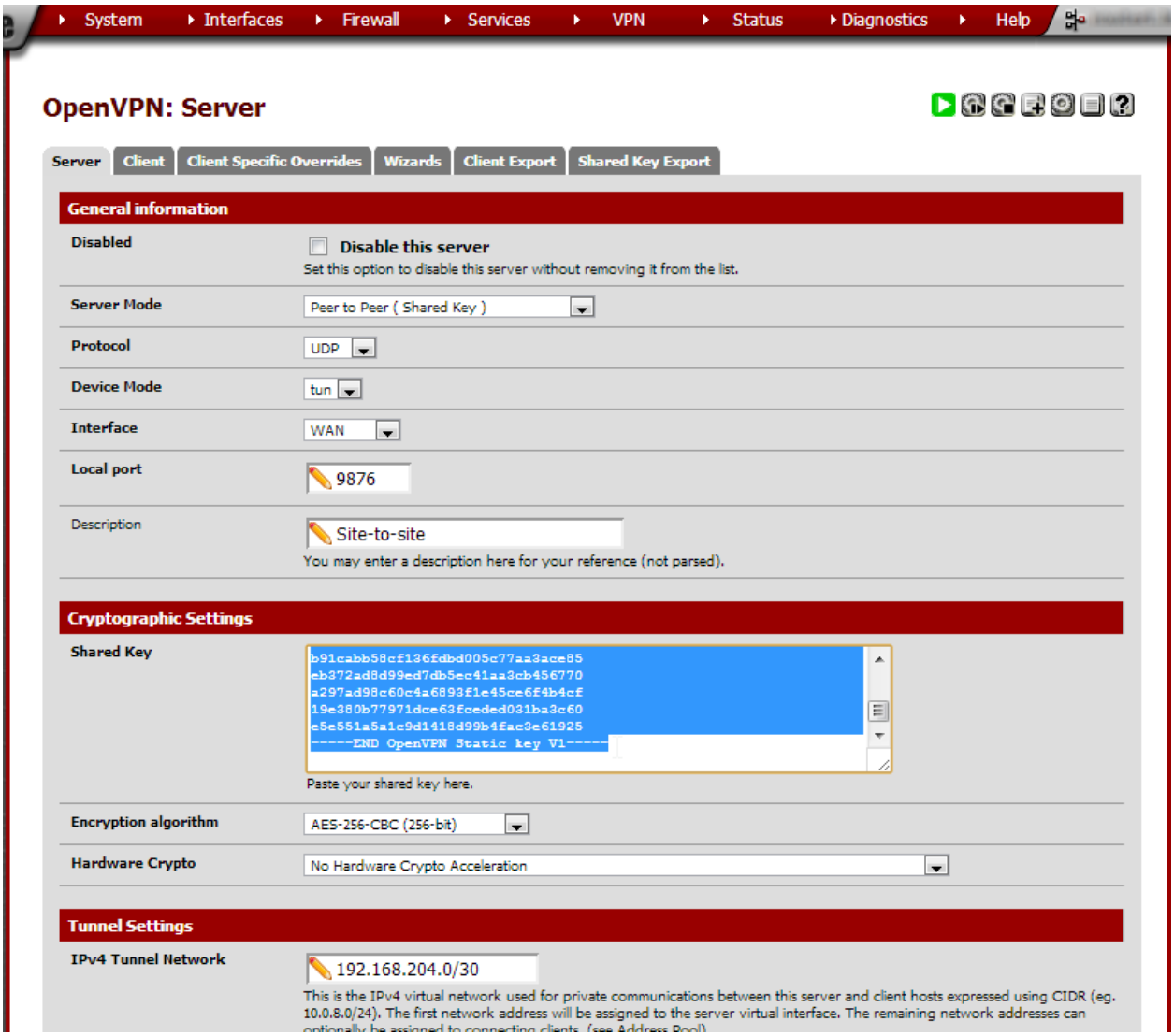

System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
OpenVPN: Server
Server
Client
Client Specific Overrides
Wizards
Client Export
Shared Key Export
General information
Disabled
Disable this server
Set this option to disable this server without removing it from the list.
Server Mode
Peer to Peer (Shared Key)
Protocol
UDP
Device Mode
tun
Interface
WAN
Local port
9876
Description
You may enter a description here for your reference (not parsed).
Cryptographic Settings
Shared Key
Automatically generate a shared key.
Encryption algorithm
AES-256-CBC (256-bit)
Hardware Crypto
No Hardware Crypto Acceleration
Tunnel Settings
IPv4 Tunnel Network
192.168.204.0/30
This is the IPv4 virtual network used for private communications between this server and client hosts expressed using CIDR (eg. 10.0.8.0/24). The first network address will be assigned to the server virtual interface. The remaining network addresses can optionally be assigned to connecting clients. (see Address Pool)
IPv6 Tunnel Network
This is the IPv6 virtual network used for private communications between this server and client hosts expressed using CIDR (eg. fe80::/64). The first network address will be assigned to the server virtual interface. The remaining network addresses can optionally be assigned to connecting clients. (see Address Pool)
IPv4 Local Network/s
192.168.20.0/24
These are the IPv4 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more CIDR ranges. You may leave this blank if you don't want to add a route to the local network through this tunnel on the remote machine. This is generally set to your LAN network.
IPv6 Local Network/s
These are the IPv6 networks that will be accessible from the remote endpoint. Expressed as a comma-separated list of one or more IP/PREFIX. You may leave this blank if you don't want to add a route to the local network through this tunnel on the remote machine. This is generally set to your LAN network.
IPv4 Remote Network/s
192.168.10.0/24
These are the IPv4 networks that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a comma-separated list of one or more CIDR ranges. If this is a site-to-site VPN, enter the remote LAN/s here. You may leave this blank if you don't want a site-to-site VPN.
IPv6 Remote Network/s
These are the IPv6 networks that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a comma-separated list of one or more IP/PREFIX. If this is a site-to-site VPN, enter the remote LAN/s here. You may leave this blank if you don't want a site-to-site VPN.
Concurrent connections
Specify the maximum number of clients allowed to concurrently connect to this server.
Compression
Compress tunnel packets using the LZO algorithm.
Type-of-Service
Set the TOS IP header value of tunnel packets to match the encapsulated packet value.
Duplicate Connections
Allow multiple concurrent connections from clients using the same Common Name.
NOTE: This is not generally recommended, but may be needed for some scenarios.
Advanced configuration
Advanced
Enter any additional options you would like to add to the OpenVPN server configuration here, separated by a semicolon
EXAMPLE: push "route 10.0.0.0 255.255.255.0";
Save

Click **Save**.

Note that our Site-to-site OpenVPN server is now shown in the Server overview. Click the **edit** button to the right of the server.

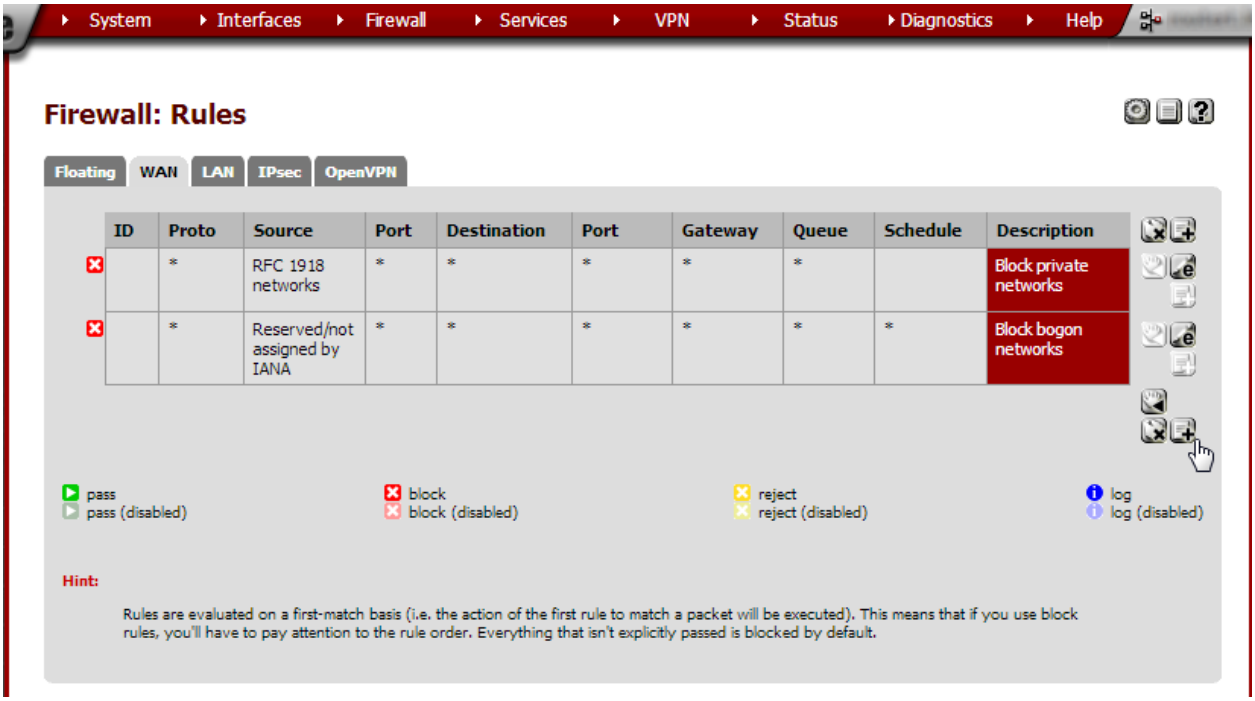


Note that in the **Cryptographic Settings** section, a Shared Key is now shown. Copy all text in the **Shared Key** text field, including the first lines beginning with # and the last line ending in —-.



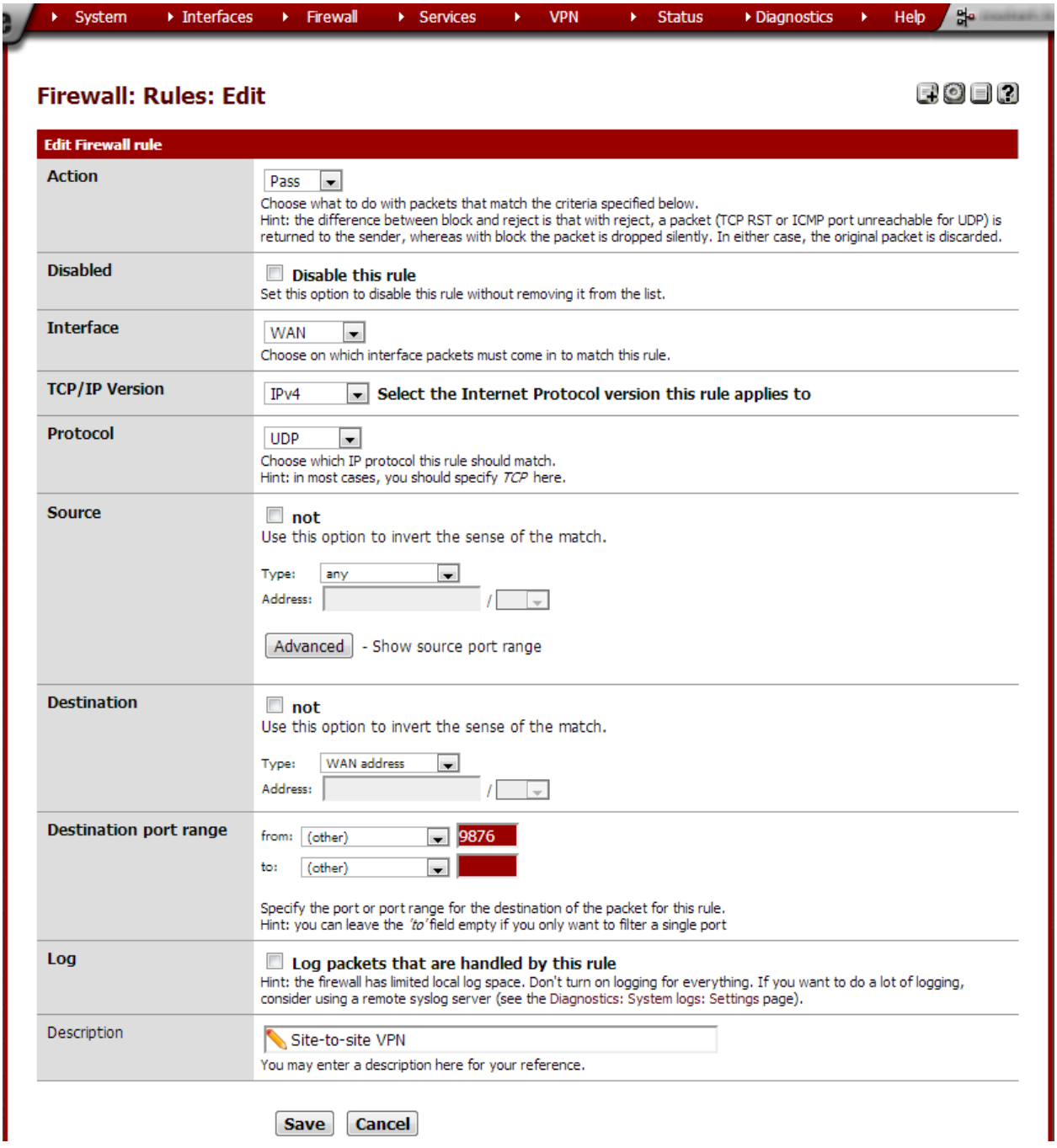
34.55.2 Configure firewall rules at Site B

From the **Firewall** menu, choose **Rules**. Open the **WAN** tab, unless using a different interface for the VPN connection. Click on the **+** button to add a new rule.

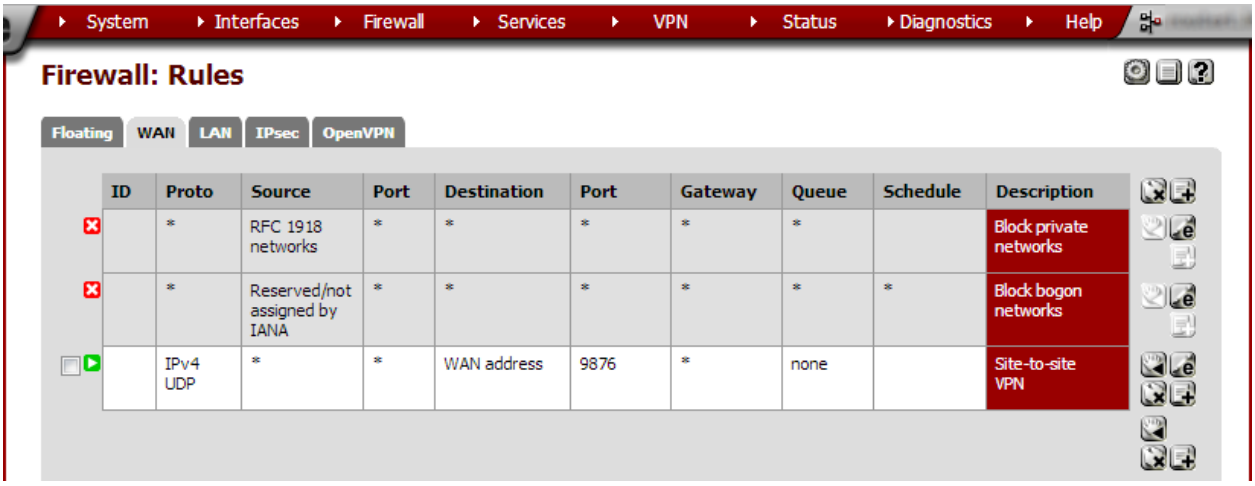


Enter these values:

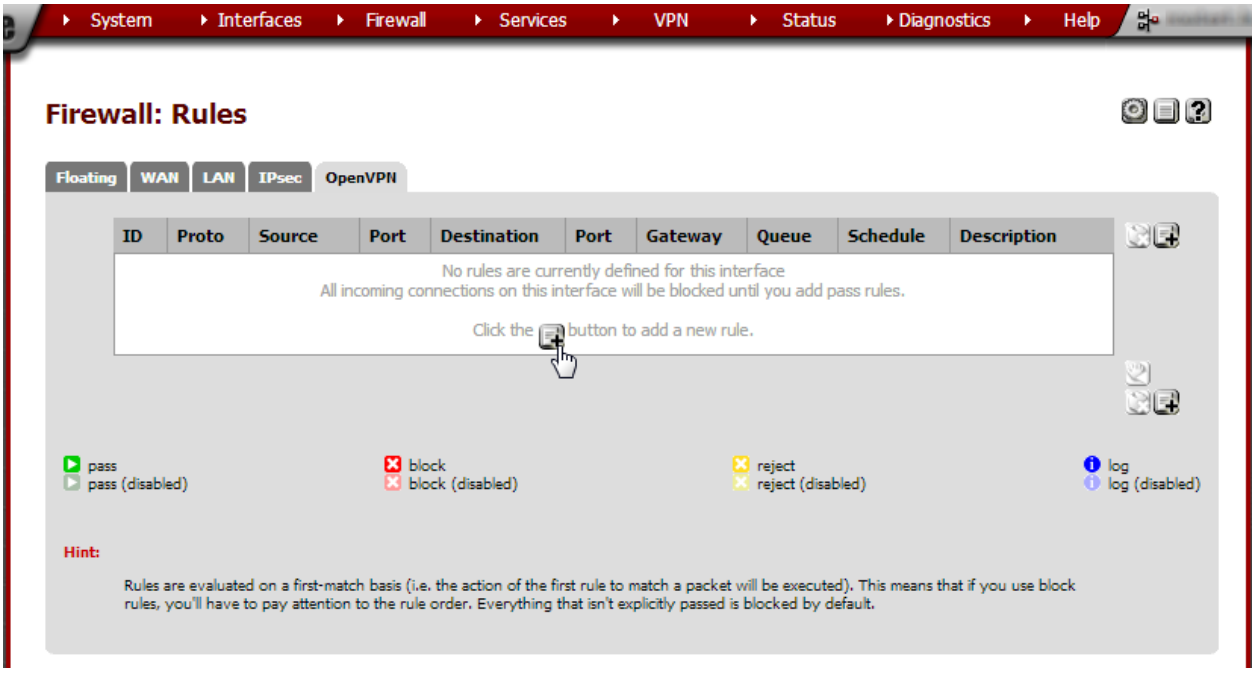
Action	Pass
Disabled	not checked
Interface	WAN
TCP/IP Version	IPv4
Protocol	UDP
Source	any
Destination	Type: WAN address
Destination port range	from: (other) 9876 to: (other)
Log	not checked
Description	Site-to-site VPN



Click **Save** and on the next page click **Apply changes**.

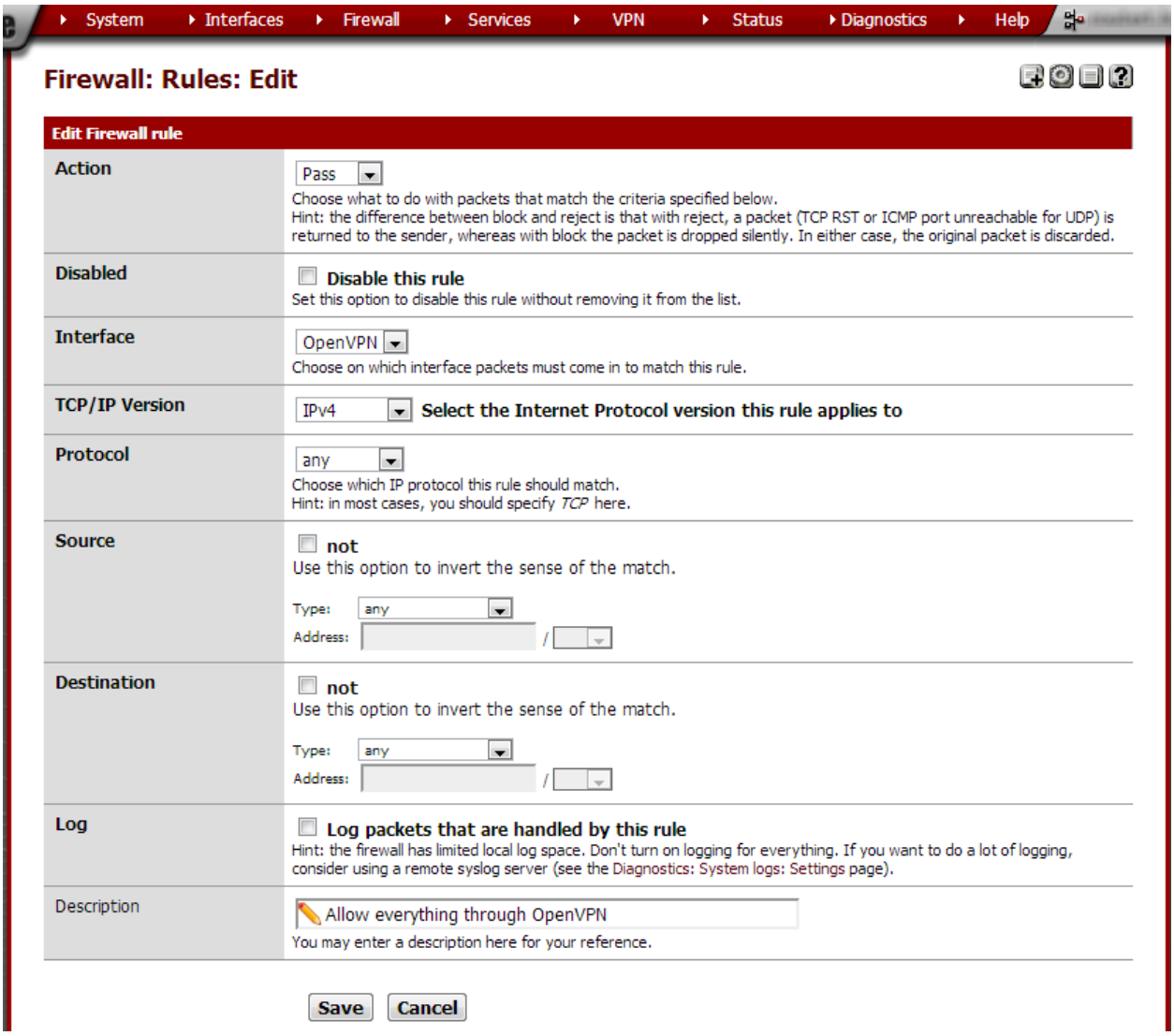


Click on the **OpenVPN** tab. We'll now add a rule to allow traffic through the OpenVPN connection. Click on the **+** button add a rule.

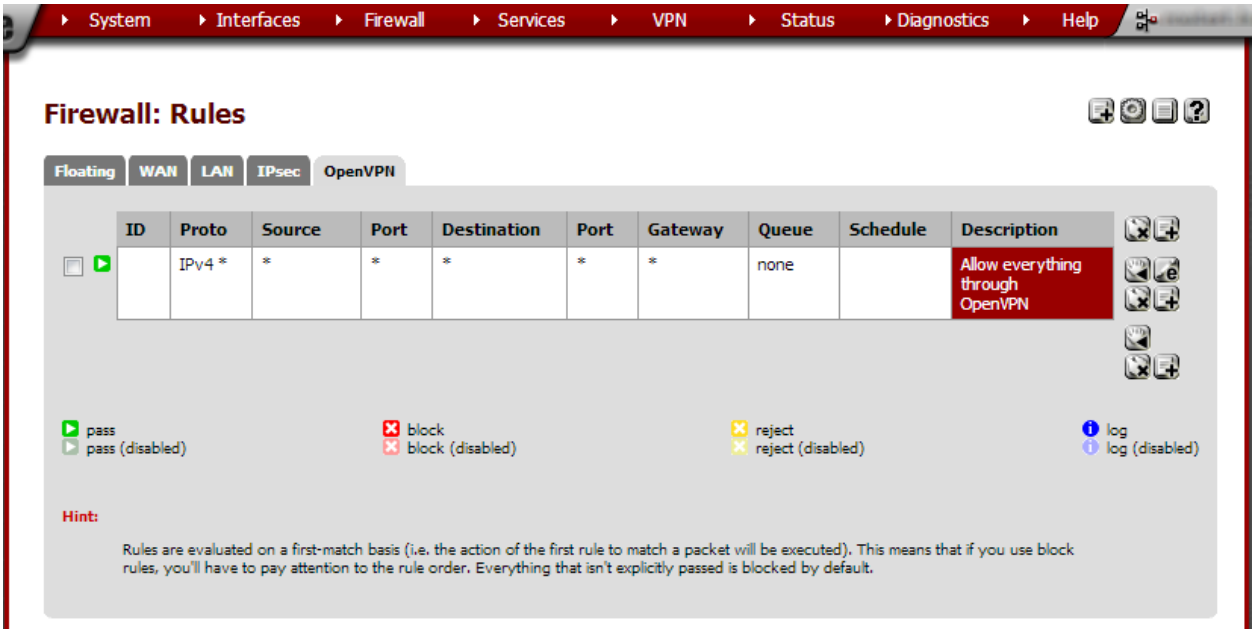


Enter these values:

Action	Pass	
Disabled	not checked	
Interface	OpenVPN	
TCP/IP Version	IPv4	
Protocol	any	
Source	any	
Destination	any	
Log	not checked	
Description	Allow everything through OpenVPN	

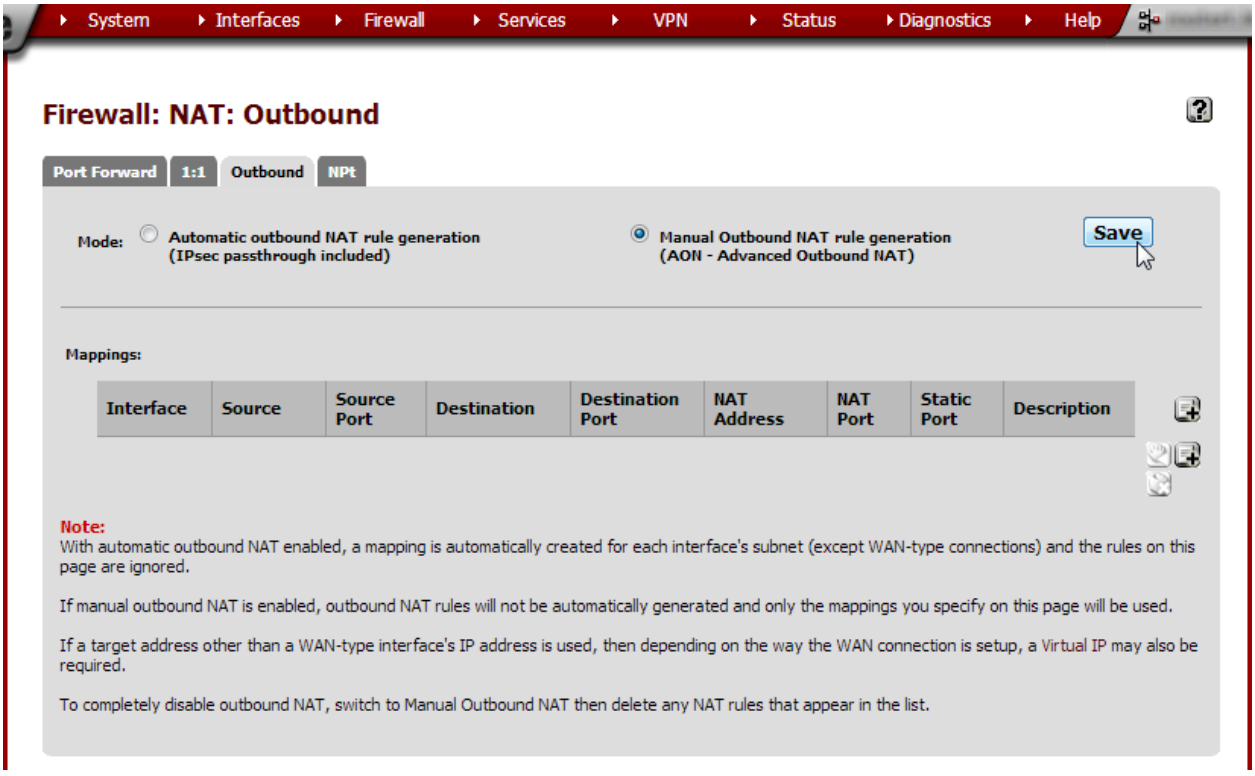


Click **Save** and on the next page **Apply Changes**.

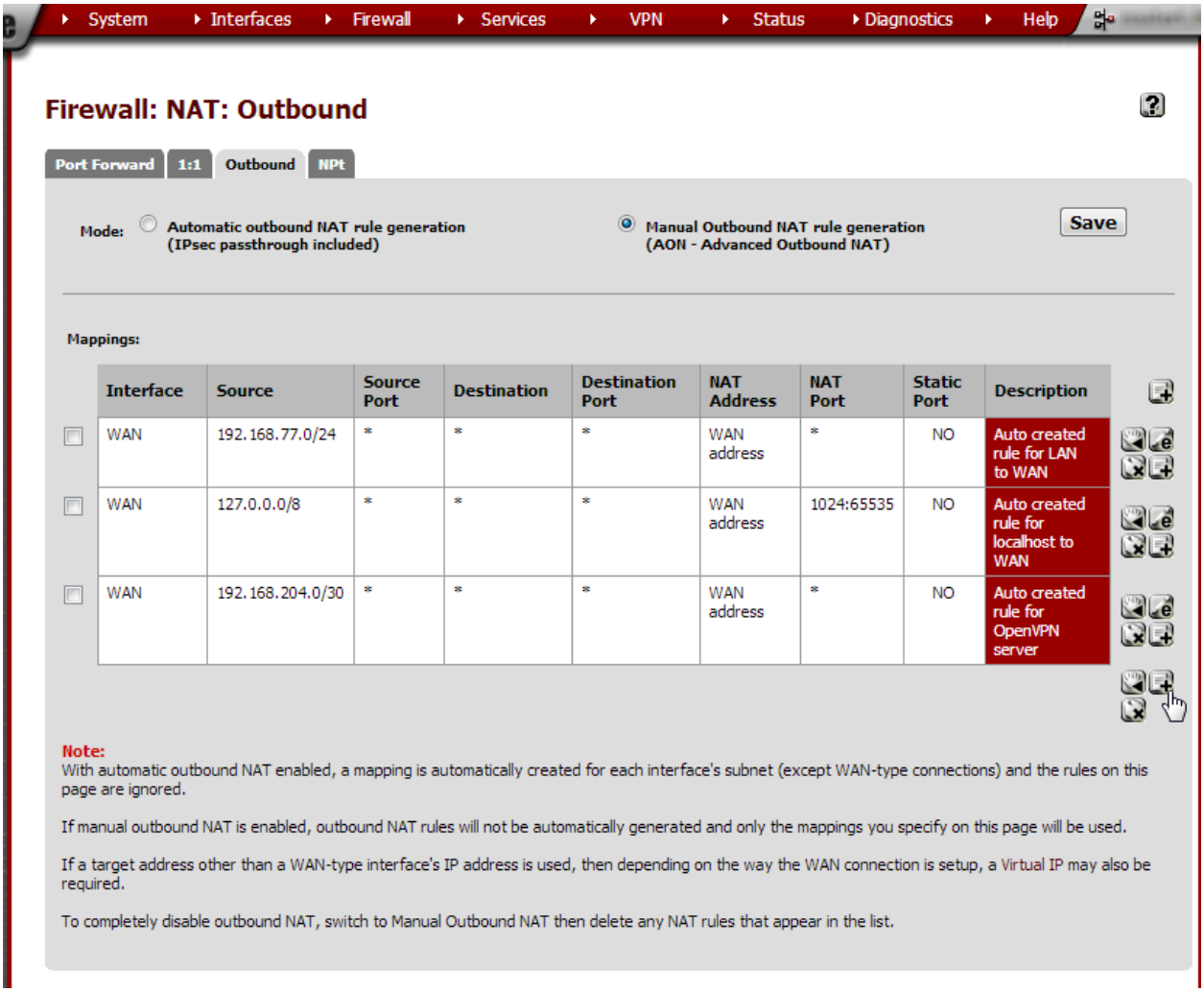


34.55.3 Set up outbound NAT at Site B

From the **Firewall** menu, choose **NAT** and click on the **Outbound** tab. Select **Manual Outbound NAT rule generation (AON – Advanced Outbound NAT)** and click **Save**. On the next page, click **Apply Changes**.

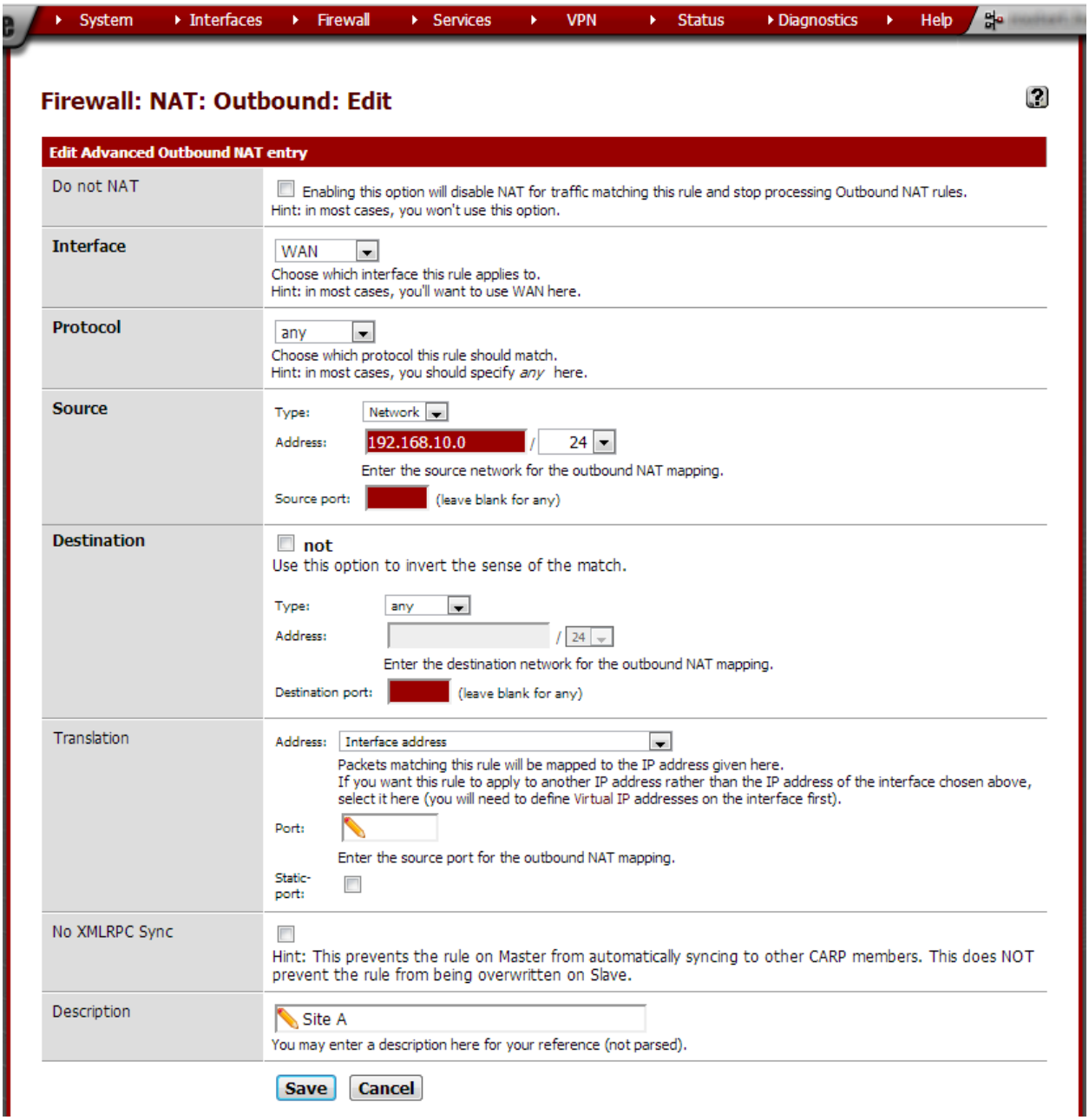


A couple of rules are generated automatically but we need to add a NAT entry for Site A's subnet. Click on the **+** button.

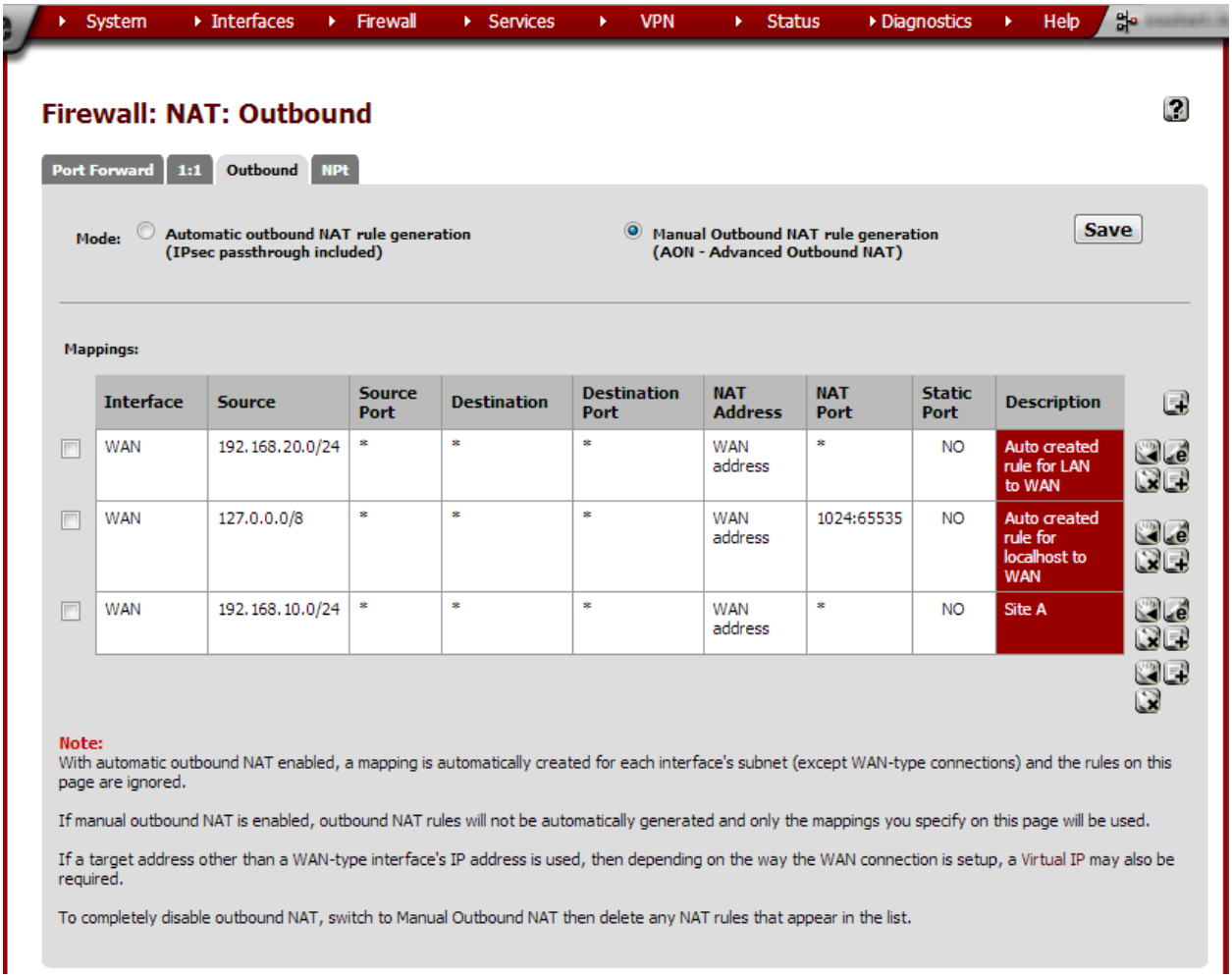


Enter these values:

Do not NAT	not checked	
Interface	WAN	Unless using a different interface for the VPN
Protocol	any	
Source	Type: Network Address: 192.168.10.0/24 Source port: leave empty	Site A's subnet
Destination	Type: any Destination port: leave empty	
Translation	Address: Interface address Port: leave empty Static port: not checked	
No XMLRPC Sync	Leave unchecked	
Description	Site A	



Click **Save** and on the next page click **Apply Changes**.



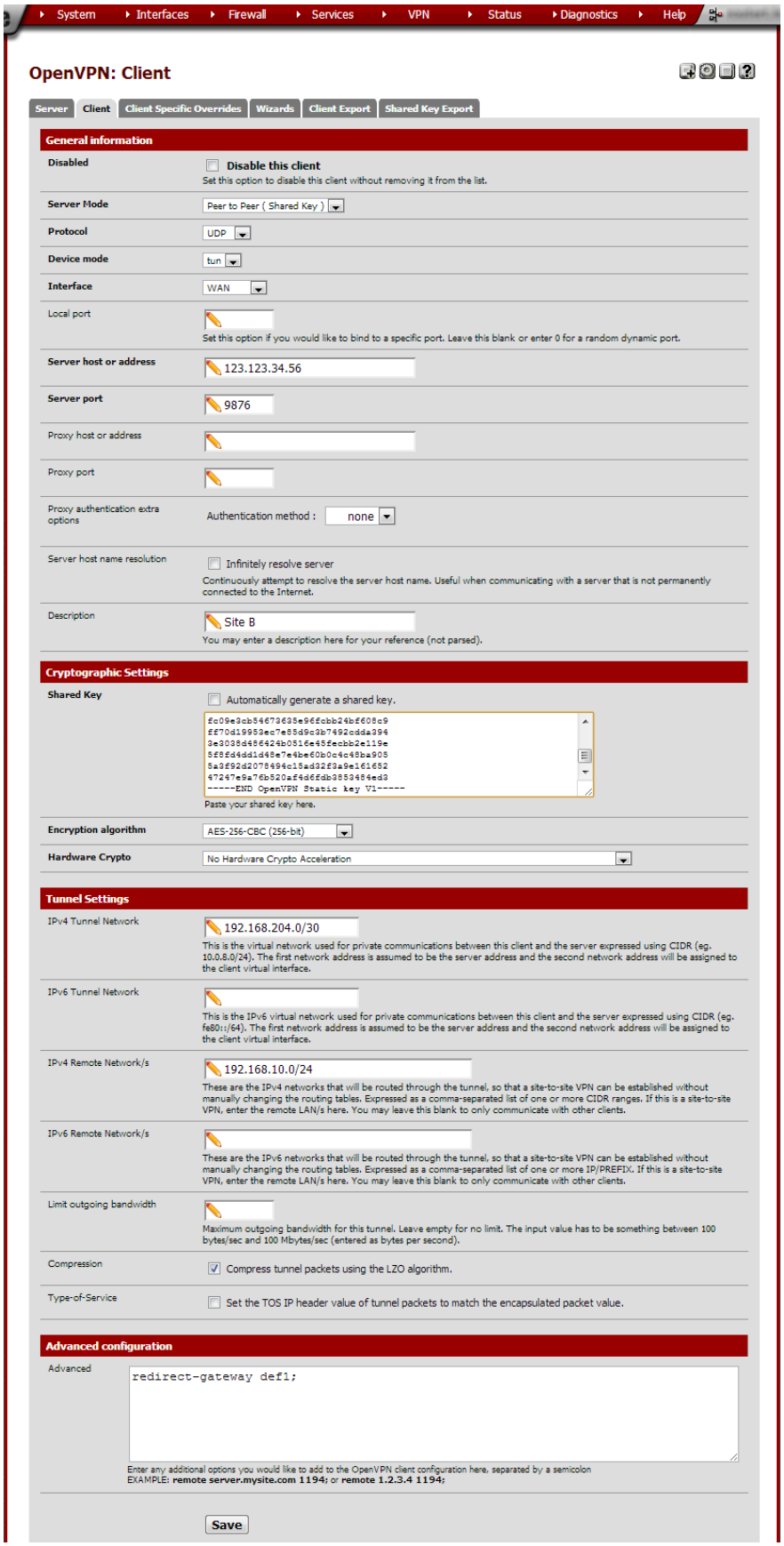
34.55.4 Set up the client at site A

From the **VPN** menu choose **OpenVPN** and go to the **Client** tab. Click the **+** button to configure a client.



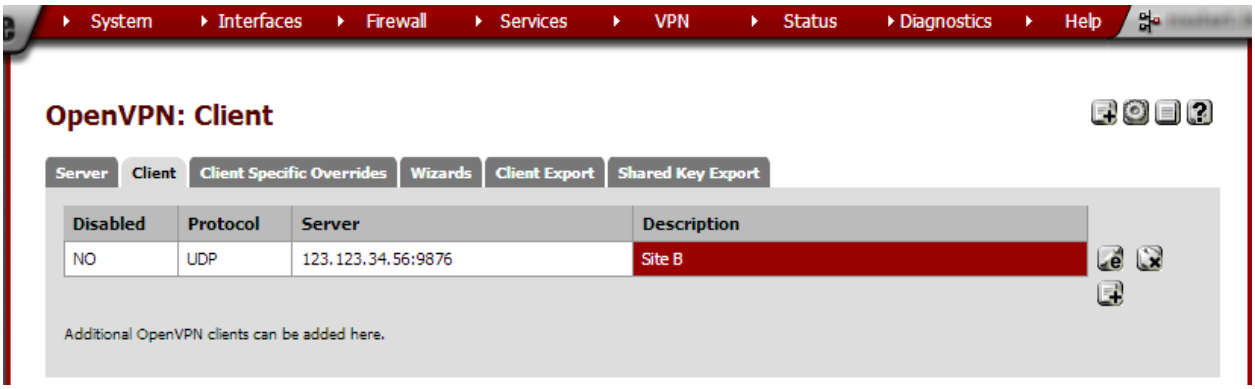
Enter these values:

Disabled	not checked	
Server Mode	Peer to Peer (Shared Key)	
Protocol	UDP	same as Site B
Device mode	tun	
Interface	WAN	
Local port	leave empty	
Server host or address	Site B's public IP address or FQDN	
Server port	9876	the port Site B is running the OpenVPN server on
Proxy host or address	leave empty if not using a proxy	
Proxy port	leave empty if not using a proxy	
Proxy authentication extra options	leave empty if not using a proxy	
Server host name resolution	check if Site B sometimes has connectivity problems	
Shared Key	do not check 'Automatically generate a shared key' but paste the Shared Key from site B	
Encryption algorithm	AES-256-CBC (256-bit)	same as Site B
Hardware Crypto	Choose 'No Hardware Crypto Acceleration' unless the hardware has an accelerator	
IPv4 Tunnel Network	192.168.204.0/30	same as Site B
IPv6 Tunnel Network	leave empty	
IPv4 Remote Network/s	192.168.20.0/24	site A's subnet
IPv6 Remote Network/s	leave empty	
Limit outgoing bandwidth	leave empty unless required	
Compression	same as Site B	
Type-of-Service	not checked	
Advanced	redirect-gateway def1;	This makes all traffic, including Internet traffic, go through the tunnel.


System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
OpenVPN: Client
Server
Client
Client Specific Overrides
Wizards
Client Export
Shared Key Export
General information
Disabled
Disable this client
Set this option to disable this client without removing it from the list.
Server Mode
Peer to Peer (Shared Key)
Protocol
UDP
Device mode
tun
Interface
WAN
Local port
Set this option if you would like to bind to a specific port. Leave this blank or enter 0 for a random dynamic port.
Server host or address
123.123.34.56
Server port
9876
Proxy host or address
Proxy port
Proxy authentication extra options
Authentication method :
none
Server host name resolution
Infinitely resolve server
Continuously attempt to resolve the server host name. Useful when communicating with a server that is not permanently connected to the Internet.
Description
Site B
You may enter a description here for your reference (not parsed).
Cryptographic Settings
Shared Key
Automatically generate a shared key.
fc09e3cb54673635e96fcbb24bf608c9
ff70d19953ec7e85d9c3b7492cdda394
3e3038d486424b0516e45fecbb2e119e
5f8fd4dd1d45e7e4be60b0c4c45ba905
5a3f92d2078494c15ad92f3a9e161652
47247e9a76b520af4d6fdb3853484ed3
-----END OpenVPN Static key V1-----
Paste your shared key here.
Encryption algorithm
AES-256-CBC (256-bit)
Hardware Crypto
No Hardware Crypto Acceleration
Tunnel Settings
IPv4 Tunnel Network
192.168.204.0/30
This is the virtual network used for private communications between this client and the server expressed using CIDR (eg. 10.0.8.0/24). The first network address is assumed to be the server address and the second network address will be assigned to the client virtual interface.
IPv6 Tunnel Network
This is the IPv6 virtual network used for private communications between this client and the server expressed using CIDR (eg. fe80::/64). The first network address is assumed to be the server address and the second network address will be assigned to the client virtual interface.
IPv4 Remote Network/s
192.168.10.0/24
These are the IPv4 networks that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a comma-separated list of one or more CIDR ranges. If this is a site-to-site VPN, enter the remote LAN/s here. You may leave this blank to only communicate with other clients.
IPv6 Remote Network/s
These are the IPv6 networks that will be routed through the tunnel, so that a site-to-site VPN can be established without manually changing the routing tables. Expressed as a comma-separated list of one or more IP/PREFIX. If this is a site-to-site VPN, enter the remote LAN/s here. You may leave this blank to only communicate with other clients.
Limit outgoing bandwidth
Maximum outgoing bandwidth for this tunnel. Leave empty for no limit. The input value has to be something between 100 bytes/sec and 100 Mbytes/sec (entered as bytes per second).
Compression
Compress tunnel packets using the LZO algorithm.
Type-of-Service
Set the TOS IP header value of tunnel packets to match the encapsulated packet value.
Advanced configuration
Advanced
redirect-gateway def1;
Enter any additional options you would like to add to the OpenVPN client configuration here, separated by a semicolon
EXAMPLE: remote server.mysite.com 1194; or remote 1.2.3.4 1194;
Save

Click **Save**.

The tunnel should now work and internet traffic should be routed from Site A through the tunnel out site B.



By Vorkbaard, 2013-07-29 - gmail{a}vorkbaard[.]nl

34.56 OpenVPN Site-to-Site Configuration Example with SSL/TLS

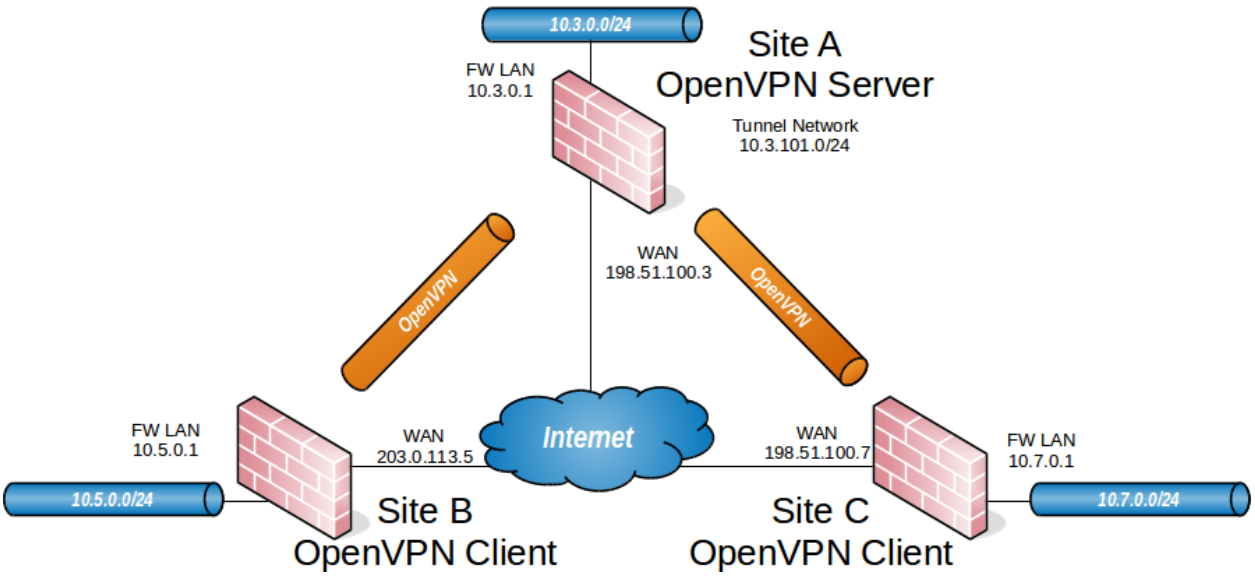


Fig. 81: OpenVPN Example Site-to-Site SSL/TLS Network

The process of configuring a site-to-site connection using **SSL/TLS** is more complicated than **Shared Key**. However, this method is typically much more convenient for managing a large number of remote sites connecting back to a central site in a hub-and-spoke fashion. It can be used for a site-to-site between two nodes, but given the increased configuration complexity, most people prefer to use shared key rather than SSL/TLS for that scenario.

When configuring a site-to-site OpenVPN connection using SSL/TLS one firewall will be the server and the others will be clients. Usually the main location will be the server side and the remote offices will act as clients, though if one location has a static IP address and more bandwidth than the main office that may be a more desirable location for the server. In addition to the subnets on both ends there will be a dedicated subnet in use for the OpenVPN interconnection between networks. This example configuration is depicted in Figure *OpenVPN Example Site-to-Site SSL/TLS Network*.

`10.3.101.0/24` is used as the IPv4 VPN Tunnel Network. The way OpenVPN allocates IP addresses is the same as for remote access clients. When using a **Topology** style of *subnet*, each client will obtain one IP address in a common subnet. When using a **Topology** style of *net30*, each connecting client gets a /30 subnet to interconnect itself with the server. See *Topology* for more details. The following sections describe how to configure the server and client sides of the connection. Any subnet can be used for this so long as it does not overlap any other subnet currently in use on the network.

In order for the server to reach the client networks behind each connection, two items are **required**:

- A *route* to tell the operating system that OpenVPN knows about a remote network
- An *iroute* in a Client-Specific Override that tells OpenVPN how to map that subnet to a specific certificate

More detail on this will follow in the example.

34.56.1 Configuring SSL/TLS Server Side

Before the VPN can be configured, a certificate structure for this VPN is required. Create a CA unique to this VPN and from that CA create a server certificate, and then a user certificate for each remote site. For the client sites, use a CN that identifies them uniquely in some way, such as their fully qualified domain name or a shortened site or hostname. For the specifics of creating a CA and Certificates, see *Certificate Management*. For this example, the CA will be called `S2SCA`, the Server CN will be `serverA`, the clients will be `clientB` and `clientC`.

- Navigate to **VPN > OpenVPN**, **Servers** tab
- Click  **Add** to create a new server
- Fill in the fields as described below, with everything else left at defaults. These options are discussed in detail earlier in the chapter. Use values appropriate for this network, or the defaults if unsure.

  Server Mode Select *Peer to Peer (SSL/TLS)*

  Protocol Select *UDP*

  Device Mode Select *tun*

  Interface Select *WAN*

  Local Port Enter `1194` unless there is another active OpenVPN server, in which case use a different port

  Description Enter text here to describe the connection

  TLS Authentication Check this box to also do TLS authentication as well as SSL

  Peer Certificate Authority Select the CA created at the beginning of this process

  Peer Certificate Revocation List If a CRL was created, select it here

  Server Certificate Select the server certificate created at the beginning of this process

  IPv4 Tunnel Network Enter the chosen tunnel network, `10.3.101.0/24`

  IPv4 Local Network Enter the LAN networks for all sites including the server: `10.3.0.0/24, 10.5.0.0/24, 10.7.0.0/24`

Note: If there are more networks on the server side that need to be reached by the clients, such as networks reachable via static routes, other VPNs, and so on, add them as additional entries in the **IPv4 Local Network** box.

  IPv4 Remote Network Enter only the client LAN networks: `10.5.0.0/24, 10.7.0.0/24`

- Click Save.

- Click  to edit the new server instance
- Find the **TLS Authentication** box
- Select all of the text inside
- Copy the text to the clipboard
- Save this to a file or paste it into a text editor such as Notepad temporarily

Next, add a firewall rule on WAN allowing access to the OpenVPN server.

- Navigate to **Firewall > Rules**, **WAN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *UDP*
- Leave the **Source** set to *any* since multiple sites will need to connect. Alternately, an alias can be made which contains the IP addresses of each remote site if they have static addresses.
- Set the **Destination** to *WAN Address*
- Set the **Destination port** to `1194` in this instance
- Enter a **Description**, such as `OpenVPN Multi-Site VPN`
- Click **Save**
- Click **Apply Changes**

A rule must also be added to the **OpenVPN** interface to pass traffic over the VPN from the Client-side LAN to the Server-side LAN. An "Allow all" style rule may be used, or a set of stricter rules. In this example allowing all traffic is OK so the following rule is made:

- Navigate to **Firewall > Rules**, **OpenVPN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *any*
- Enter a **Description** such as `Allow all on OpenVPN`
- Click **Save**
- Click **Apply Changes**

The last piece of the puzzle is to add **Client Specific Overrides** for each client site. These are needed to tie a client subnet to a particular certificate for a site so that it may be properly routed.

- Navigate to **VPN > OpenVPN**, **Client Specific Overrides** tab
- Click  to add a new override
- Fill in the fields on this screen as follows:

  Common Name Enter the CN of the first client site. In this example, that is `clientB`.

  IPv4 Remote Network This field sets up the required *iroute* so enter the clientB LAN subnet, `10.5.0.0/24`
- Click **Save**

Add an override for the second site, adjusting the **Common Name** and **IPv4 Remote Network** as needed. In the example for site C, these values would be `clientC` and `10.7.0.0/24` respectively.

The next task is to export the certificates and keys needed for clients.

- Navigate to **System > Cert Manager**
- Click the links to export the following items:
  - CA Certificate
  - Client site certificate (`.crt`) for each client location.
  - Client site key (`.key`) for each client location.

Warning: *Do not* export the CA key, server certificate, or server key. They are not needed on the clients, and copying them unnecessarily significantly weakens the security of the VPN.

That completes the server setup, next, now move on to configure the clients.

34.56.2 Configuring SSL/TLS Client Side

On the client, import the CA certificate along with the client certificate and key for that site. This is the same CA and client certificate made on the server and exported from there. This can be done under **System > Cert Manager**. For specifics on importing the CA and certificates, see *Certificate Management*.

After importing the certificates, create the OpenVPN client:

- Navigate to **VPN > OpenVPN**, **Client** tab
- Click  **Add** to create a new client
- Fill in the fields as follows, with everything else left at defaults

  Server Mode Select *Peer to Peer (SSL/TLS)*

  Protocol Select *UDP*

  Device Mode Select *tun*

  Interface Select *WAN*

  Server host or address Enter the public IP address or hostname of the OpenVPN server here (e.g. `198.51.100.3`)

  Server Port Enter `1194` or whichever port was configured on the server

  Description Enter text here to describe the connection

  TLS Authentication Check **Enable authentication of TLS packets**, Uncheck **Automatically generate a shared TLS authentication key**, then paste in the TLS key for the connection here using the key copied from the server instance created previously

  Peer Certificate Authority Select the CA imported at the beginning of this process

  Client Certificate Select the client certificate imported at the beginning of this process

- Click Save

A rule must also be added to the **OpenVPN** interface to pass traffic over the VPN from the Client-side LAN to the Server-side LAN. An "Allow all" style rule may be used, or a set of stricter rules. In this example allowing all traffic is OK so the following rule is made:

- Navigate to **Firewall > Rules**, **OpenVPN** tab
- Click  **Add** to create a new rule at the top of the list
- Set **Protocol** to *any*
- Enter a **Description** such as `Allow all on OpenVPN`
- Click **Save**
- Click **Apply Changes**

The configuration of the client is complete. No firewall rules are required on the client WAN because the client only initiates outbound connections.

Note: With remote access PKI configurations, routes and other configuration options are not usually defined in the client configuration but rather they are pushed from the server to the client. If there are more networks to reach on the server side, configure them on the server to be pushed.

34.56.3 Testing the connection

The configuration is now complete and the connection will immediately be active upon saving on the client side. Try to ping across to the remote end to verify connectivity. If problems arise, refer to *Troubleshooting OpenVPN*.

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

34.57 WireGuard Remote Access VPN Configuration Example

This recipe covers configuring a basic *WireGuard* remote access style VPN tunnel.

Note: Though WireGuard does not have a concept of "Client" and "Server" per se, in this style of deployment the firewall cannot initiate connections to remote peers. In this way the firewall acts like a "Server" and may be referred to as such in this documentation. Remote peers may also be referred to as "clients".

34.57.1 Required Information

The following basic information must be determined before starting the VPN configuration.

Item	Value
Design	Remote access, one tunnel+many peers
Firewall WAN	`198.51.100.6`
Listen Port	`51820`
Tunnel Subnet	`10.6.210.0/24`
Tunnel Address	`10.6.210.1/24`
Peer Addresses	`10.6.210.2 - 10.6.210.254`
Peer Endpoints	Dynamic

Generating Keys

WireGuard requires public/private key pairs for each peer, including this firewall.

Warning: Keys cannot be reused between clients, as WireGuard requires unique keys to identify clients and were to send their traffic.

Tunnel Keys

To generate keys for the firewall itself, click the **Generate** button when configuring a tunnel. The GUI will populate the private and public key fields automatically.

The peers will need the public key for their configuration.

Peer Keys

Each peer will need its own public/private key pair. The private key will be needed on the peer client software while the public key will be needed on the firewall itself for the peer definition.

These keys can be generated by the clients themselves, or via command line on a system which has the WireGuard utilities installed. This includes the firewall itself; these commands may be run from a console or SSH shell or from **Diagnostics > Command Prompt**.

From a command line, execute the following:

```
$ wg genkey | tee privatekey | wg pubkey > publickey
```

This command outputs files named `privatekey` and `publickey` which respectively contain a private key and its associated public key. This key pair can be used for a WireGuard peer.

To view the keys, inspect the contents of the files:

```
$ cat privatekey
WGpL3/ejM5L9ngLoAtXkSP1QTNp4eSD34Zh6/Jfni1Q=
$ cat publickey
b9FjbupGC7fomO5U4jL5Irt1ZV5rq4c+utGKj53HXgU=
```

Repeat the commands as needed as many times as is necessary for the number of peers required by this tunnel. Note the keys in a secure place.

Tip: Change the commands to output files named for their associated peer, then store the resulting files in a secure location.

Alternately, the keys can be output in one command without storing them persistently. This behavior is not be supported on all platforms, but is supported on the firewall itself.

```
$ wg genkey | tee /dev/stderr | wg pubkey
4BSH81zC3/OWl25XrzqWy7WnAiARXySHd+K+KFxNrWU=
rzWOC0zH9v2zF6r92uCbjs7JOmhqy8N+cUdA+GCynSM=
```

34.57.2 Tunnel Configuration

Now it's time to create the WireGuard tunnel.

- Navigate to **VPN > WireGuard**
- Click  **Add Tunnel**
- Fill in the options using the information determined earlier:

  Enabled *Checked*

  Description `Remote Access`

  Address `10.6.210.1/24`

  Listen Port `51820`

  Interface Keys Click **Generate** to create a new set of keys.
- Click **Save**, or continue on to add the peers

34.57.3 Peer Configuration

Peers can be added when editing a tunnel. To edit a tunnel:

- Navigate to **VPN > WireGuard**
- Locate the WireGuard tunnel to which the peers will be added
- Click  at the end of the row for the tunnel

From the tunnel editing page, add a peer as follows:

- Click  **Add Peer**
- Fill in the options using the information determined earlier:

  Description The name of this client (e.g. The name of a person, device, username, or other uniquely identifying information.)

  Endpoint Leave blank for dynamic endpoints.

Endpoint Port Leave blank for dynamic endpoints.

Keepalive Typically left blank, but may be filled in if clients have problems traversing certain firewalls.

Public Key The public key for this peer. Obtained from the key generation process earlier, or from the peer itself if it was generated by client software directly.

Allowed IPs The tunnel IP address for this peer, from the list determined above, with a `/32` CIDR mask. For example, the first peer will be `10.6.210.2/32`, the second will be `10.6.210.3/32`, and so on.

Peer WireGuard Address The IP address assigned to the client again. This field can differ from **Allowed IPs** in many cases, but in this type of configuration it is generally the same, as remote access clients do not usually have additional networks routed to them. Use the same subnet mask as the **Tunnel** address, rather than `/32`.

Pre-Shared Key Not used in this example, but for additional security this pre-shared key can be generated and copied to the peer. Must match on the client and server.

- Click **Update**
- Repeat the steps to add additional peers as needed.
- Click **Save**

34.57.4 Firewall Rules

First add a rule to pass external WireGuard traffic on the WAN:

- Navigate to **Firewall > Rules**, **WAN** tab
- Click  **Add** to add a new rule to the top of the list
- Use the following settings:

  Action *Pass*

  Interface *WAN*

  Protocol *UDP*

  Source *any*

  Destination *WAN Address*

  Destination Port Range *(other)*, `51820`

  Description `Pass traffic to WireGuard`

- Click **Save**
- Click **Apply Changes**

Next, add a rule to pass traffic inside the WireGuard tunnel:

- Navigate to **Firewall > Rules**, **WireGuard** tab
- Click  **Add** to add a new rule to the top of the list
- Use the following settings:

  Action *Pass*

Interface *WireGuard*

Protocol *Any*

Source *any*

Destination *any*

Description `Pass VPN traffic from WireGuard peers`

- Click **Save**
- Click **Apply Changes**

34.57.5 Client Configuration

Client configuration varies by platform, see WireGuard documentation for details. This section covers a basic configuration.

This is an example configuration from a WireGuard client, such as one found on Windows or Android:

```
[Interface]
PrivateKey = WGpL3/ejM5L9ngLoAtXkSP1QTNp4eSD34Zh6/Jfni1Q=
ListenPort = 51820
Address = 10.6.210.2/24

[Peer]
PublicKey = PUVBJ+zuz/0mRPEB4tIaVbet5NzVwdWMX7crGx+/wDs=
AllowedIPs = 10.6.210.1/32, 10.6.0.0/24
Endpoint = 198.51.100.6:51820
```

The fields in that file are as follows:

Interface Settings for this client.

PrivateKey The private key for this peer. Obtained from the key generation process earlier, or from the peer itself if it was generated by client software directly.

ListenPort A static port to listen on, or omit the line to use a random port instead.

Address The tunnel address for this client. Not supported on all platforms, as some require configuring the address using command-line utilities. However, clients on Windows and Android, for example, support this directive.

This should use the same CIDR mask as the **Tunnel** address. In this example, the first peer is `10.6.210.2/24`.

Peer Configuration for the firewall end of the tunnel.

PublicKey The public key from the **Tunnel** configuration on the firewall.

AllowedIPs The **Tunnel** address, and any additional networks which should be routed across the VPN in a comma-separated list. This could be a LAN subnet (e.g. `10.6.0.0/24`) or use `0.0.0.0/0` to route all traffic, including Internet traffic, across the tunnel.

Endpoint The firewall WAN IP address and WireGuard **Listen Port**

This only covers the basics, there are numerous other fields which can be used to control client behavior plus additional client options which vary by platform. For additional details, see the WireGuard documentation and the documentation for the WireGuard software used by a peer.

Transfer the resulting client configuration file to the peer in a secure manner. Methods vary by platform and client software.

34.57.6 Finish Up

After configuring the client and activating the VPN, the client should be able to pass traffic to the networks listed in the `AllowedIPs` list in its configuration.

See also:

- *WireGuard*
- *WireGuard Site-to-Site VPN Configuration Example*
- *WireGuard VPN Client Configuration Example*

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

34.58 WireGuard Site-to-Site VPN Configuration Example

This recipe explains how to setup a VPN tunnel between two firewalls using *WireGuard*.

This example is a minimal configuration, more complicated scenarios are possible, see *WireGuard* for details.

34.58.1 Required Information

General Values

Item	Value
Design	Site-to-Site, one peer per tunnel
Tunnel Subnet	`10.6.210.0/24`

HQ:

Item	Value
WAN IP Address	`198.51.100.15`
Tunnel Address	`10.15.210.1/24`
Listen Port	`51820`
LAN Subnet	`10.15.0.0/24`

Satellite Office:

Item	Value
WAN IP Address	`198.51.100.23`
Tunnel Address	`10.15.210.2/24`
Listen Port	`51820`
LAN Subnet	`10.23.0.0/24`

34.58.2 Tunnel Configuration

First create the WireGuard tunnel on both sites:

- Navigate to **VPN > WireGuard**
- Click  **Add Tunnel**
- Fill in the options using the information determined earlier, with variations noted for each site:

  Enabled *Checked*

  HQ Settings

  Description `Satellite Office VPN`

  Address `10.15.210.1/24`

  Satellite Office Settings

  Description `HQ VPN`

  Address `10.15.210.2/24`

  Listen Port `51820`

  Interface Keys Click **Generate** to create a new set of keys.
- Copy the public key from each firewall and note which is which
- Click **Save**

34.58.3 Peer Configuration

The peer entry for the server can be added when editing the tunnel. Follow these steps on both sites, with the differences in settings noted inline.

Edit the tunnel:

- Navigate to **VPN > WireGuard**
- Locate the WireGuard tunnel for this VPN provider
- Click  at the end of the row for the tunnel

From the tunnel editing page, add a peer:

- Click  **Add Peer**
- Fill in the options using the information determined earlier:

  HQ Settings

Description `Satellite Office Peer`

Endpoint `198.51.100.23` (the WAN IP address of the Satellite Office)

Endpoint Port `51820`

Public Key The public key *from the Satellite Office firewall*

Allowed IPs `10.15.210.2/32, 10.23.0.0/24` (Satellite office tunnel address and LAN)

Peer WireGuard Address `10.15.210.2` (Satellite office tunnel address)

Satellite Office Settings

Description `HQ VPN Peer`

Endpoint `198.51.100.15` (the WAN IP address of HQ)

Endpoint Port `51820`

Public Key The public key *from the HQ firewall*

Allowed IPs `10.15.210.1/32, 10.15.0.0/24` (HQ tunnel address and LAN)

Peer WireGuard Address `10.15.210.1` (HQ tunnel address)

- Click **Update**
- Click **Save**

34.58.4 Assign Interface

These steps should be done on both sites.

First, fix the default gateway so WireGuard isn't automatically selected before it's ready:

- Navigate to **System > Routing**
- Set **Default Gateway IPv4** to a specific gateway (e.g. *WANGW*) or group
- Set **Default Gateway IPv6** in a similar manner if this VPN will also carry IPv6 traffic
- Click **Save**
- Click **Apply Changes**

Next, assign the interface so the automatic gateway can be used for policy routing (*Assign a WireGuard Interface*).

- Navigate to **Interfaces > Assignments**
- Select the appropriate `wg<number>` interface in the **Available network ports** list
- Click  **Add** to assign the interface as a new OPT interface (e.g. OPT1)
- Navigate to the Interface configuration page, **Interfaces > OPTx**
- Check **Enable**
- Enter an appropriate **Description** which will become the interface name (e.g. `VPN_HQ` or `VPN_Satellite`)
- Click **Save**
- Click **Apply Changes**

34.58.5 Firewall Rules

First, add a rule to the WAN on both firewalls to allow traffic to reach WireGuard:

- Navigate to **Firewall > Rules**, **WAN** tab
- Click  **Add** to create a new firewall rule at the top of the list so that it matches before other rules
- Configure the firewall rule as follows:

  Action *Pass*

  Protocol *UDP*

  Source This can typically be left at *Any*, but it is more secure to fill in the IP address of the opposing firewall.

  Destination *WAN Address*

  Destination Port Range *(other)*, `51820`

  Description `Pass traffic to WireGuard`

- Click **Save**
- Click **Apply Changes**

Next, add a rule to pass traffic inside the WireGuard tunnel on both firewalls:

- Navigate to **Firewall > Rules**
- Click the tab for the assigned WireGuard interface (e.g. **VPN_Satellite** or **VPN_HQ**)
- Click  **Add** to add a new rule to the top of the list
- Use the following settings:

  Action *Pass*

  Protocol *Any*

  Source *any*

  Destination *any*

  Description `Pass VPN traffic from WireGuard peers`

  Note: This rule allows all traffic between sites, which is easy but not a secure practice. Traffic between the sites can be restricted as needed with less permissive rules.

- Click **Save**
- Click **Apply Changes**

34.58.6 Routing

Since the networks for the peers were entered in the **Allowed IPs** field, the firewall will automatically handle setting up routes to the peer, so no additional routing work should be needed.

See also:

As an alternative to static routing in this way, dynamic routing protocols can also work with WireGuard. See *WireGuard Routing* for more information.

The interface assignment also creates an automatic gateway which can be used for policy routing if needed.

34.58.7 Finish Up

The configuration is now complete! The two sites should now have full LAN-to-LAN connectivity.

See also:

- *WireGuard*
- *Routing*
- *WireGuard Remote Access VPN Configuration Example*
- *WireGuard VPN Client Configuration Example*

Warning: WireGuard has been removed from the base system in releases after pfSense Plus 21.02-p1 and pfSense CE 2.5.0, when it was removed from FreeBSD.

If upgrading from a version that has WireGuard active, the upgrade will abort until all WireGuard tunnels are removed. For more details, see the *Release Notes*

WireGuard is available as an experimental add-on package on pfSense Plus 21.05, pfSense CE 2.5.2, and later versions. The settings for the WireGuard add-on package are not compatible with the older base system configuration.

34.59 WireGuard VPN Client Configuration Example

This recipe explains how to setup *WireGuard* as a "client" to a remote VPN service through which Internet traffic will be routed.

Note: Though WireGuard does not have a concept of "Client" and "Server" per se, in this style of deployment the firewall initiates connections to a remote peer but the peer never initiates back to the firewall. In this way, the firewall behaves like a "Client" and may be referred to as such in this document. The remote peer may also be referred to as "server".

34.59.1 Required Information

The following basic information must be determined before starting the VPN configuration.

Item	Value
Tunnel Address	`10.6.210.2/24`
Interface Keys	Generate locally or from provider
Peer Public Key	Obtain from provider
Peer Endpoint	`198.51.100.6`
Peer Port	`51820`
Peer WG Address	`10.6.210.1`
Peer DNS Server	`10.6.0.1` (No TLS)
Allowed IPs	`0.0.0.0/0` (All traffic)

Some of these values must be obtained from the VPN provider or server administrator. Methods vary, but some may have a web-based portal which shows settings or generates a configuration file. Others may opt to send settings in a more secure manner.

Keys

In this role, the source of the keys can vary. Ideally, a private and public key set for this firewall should be generated by this firewall and the private key should never leave. The public key should be copied and submitted to the administrator of the server side so it can be used for this client.

Some providers insist on generating the keys themselves so they can preallocate addresses and other settings based on keys they already know. This also allows them to easily generate configurations for clients. It's less secure this way, but more convenient.

34.59.2 Tunnel Configuration

First create the WireGuard tunnel.

- Navigate to **VPN > WireGuard**
- Click  **Add Tunnel**
- Fill in the options using the information determined earlier:

  Enabled *Checked*

  Description `VPN Provider`

  Address `10.6.210.2/24`

  Listen Port This does not likely matter unless the server requires a specific source port. In most cases it can be left blank or at the default `51820`.

  Interface Keys Click **Generate** to create a new set of keys, or enter the private key supplied by the provider.

- Click **Save**, or continue on to add the peer

34.59.3 Peer Configuration

The peer entry for the server can be added when editing the tunnel. To edit the tunnel:

- Navigate to **VPN > WireGuard**
- Locate the WireGuard tunnel for this VPN provider
- Click  at the end of the row for the tunnel

From the tunnel editing page, add a peer as follows:

- Click  **Add Peer**
- Fill in the options using the information determined earlier:

  Description The name of this server or VPN provider

  Endpoint The server hostname or IP address, `198.51.100.6` in this example.

  Endpoint Port The server WireGuard port, `51820` in this example.

  Public Key The public key for the **VPN provider endpoint**, given by the VPN provider.

  Allowed IPs List of networks to route to the remote side. Since this example will be sending all traffic through the VPN provider, enter `0.0.0.0/0`.

  Peer WireGuard Address If known, the IP address of the server in the tunnel subnet (e.g. `10.6.210.1`).

  Pre-Shared Key Not used in this example, but for additional security this pre-shared key can be generated and copied to the peer. Must match on the client and server.
- Click **Update**
- Click **Save**

34.59.4 Assign Interface

First, fix the default gateway so WireGuard isn't automatically selected before it's ready:

- Navigate to **System > Routing**
- Set **Default Gateway IPv4** to a specific gateway (e.g. *WANGW*) or group
- Set **Default Gateway IPv6** in a similar manner if this VPN will also carry IPv6 traffic
- Click **Save**
- Click **Apply Changes**

Next, assign the interface so the automatic gateway can be used for policy routing (*Assign a WireGuard Interface*).

- Navigate to **Interfaces > Assignments**
- Select the appropriate `wg<number>` interface in the **Available network ports** list
- Click  **Add** to assign the interface as a new OPT interface (e.g. OPT1)
- Navigate to the Interface configuration page, **Interfaces > OPTx**
- Check **Enable**

- Enter an appropriate **Description** which will become the interface name (e.g. `WG_VPN`)
- Click **Save**
- Click **Apply Changes**

34.59.5 Gateways and Groups

Assigning the interface creates an automatic gateway named after the interface itself. In this example, since the interface was named `WG_VPN`, the automatic gateway is `WG_VPN_WGV4`.

This gateway can be added to a gateway group for failover or load balancing of outbound traffic. This example assumes there are no existing groups. If there are groups already, the new gateway can be added to them like any other.

This example sets up a Gateway Group which prefers WireGuard and fails over to WAN. Traffic directed to this group will use WireGuard when it is up, and WAN when it is down. In practice this specific behavior may or may not be desirable, but can be used as a template for other scenarios.

Note: This will only function properly if gateway monitoring is possible. For that to work, one of two things must be done:

- The WireGuard **Peer WireGuard Address** must be filled in with the remote end of the VPN and that address must respond to ICMP echo (ping) requests over WireGuard.
- Alternately, edit the automatic WireGuard interface gateway and fill in a different **Monitor IP** address which respond to ICMP echo (ping) requests over WireGuard.

To create a new group:

- Navigate to **System > Routing**, **Gateway Groups** tab
- Click  **Add**
- Configure the group as follows:

  Group Name `Prefer_WireGuard`

  Gateway Priority

  WG_VPN_WGV4 *Tier 1*

  WANGW *Tier 2*

  Description `Prefer VPN, fail to WAN`
- Click **Save**
- Click **Apply Changes**

34.59.6 Outbound NAT

By default the VPN will not have outbound NAT applied to its traffic. Most VPN providers will require this, so that all traffic appears to originate from the address of the VPN interface, and not LAN.

To setup outbound NAT for the VPN:

- Navigate to **Firewall > NAT**, **Outbound** tab
- Set **Mode** to **Hybrid Outbound NAT**

  This example assumes the firewall starts out on **Automatic Outbound NAT**. If the firewall is using **Manual Outbound NAT**, there is no need to change the mode.
- Click **Save**
- Click  **Add** to create a new outbound NAT rule at the top of the list
- Configure the NAT rule as follows:

  Interface The assigned WireGuard interface (e.g. *WG_VPN*)

  Source The LAN subnet of this firewall (e.g. `192.168.1.0/24`)

  Description A description of the rule, if desired: `Outbound NAT for LAN to WireGuard VPN Provider`

  Leave all remaining options at their default values
- Click **Save**
- Click **Apply Changes**

34.59.7 Firewall Rules

This scenario should not require any firewall rules on the WAN or VPN interface. No connections will be made inbound on the WAN, only outbound. Traffic from the Internet will not be allowed back into the VPN interface.

34.59.8 Routing Traffic

Policy Routing

Rules can be added to local interfaces, such as LAN, for policy routing which utilize the gateway for the WireGuard interface.

For example, to policy route all traffic from a host on the LAN out through WireGuard:

- Navigate to **Firewall > Rules**, **LAN** tab
- Click  **Add** to create a new firewall rule at the top of the list so that it matches before other rules
- Configure the firewall rule as follows:

  Action *Pass*

  Interface *LAN*

  Protocol *Any*

Source Set this to match the client whose outbound traffic will be routed across the VPN. For example:

Single host or alias, `192.168.1.23`

Destination *Any*

Gateway *WG_VPN_WGV4*

Note: Click **Display Advanced** to show this option.

Leave all remaining options at their default values

- Click **Save**
- Click **Apply Changes**

This concept can be adapted for a number of different scenarios. For example, match all LAN traffic and send it across the VPN, or match traffic and use a gateway group to prefer the VPN, etc.

Static Routing

Specific networks can be routed across the VPN statically in a couple different ways:

- Add the network(s) to the **Allowed IPs** list for the remote peer on this tunnel. This automatically adds a route to the operating system routing table.
- Add a static route for the network(s) under **System > Routing** on the **Static Routes** tab.

Default Gateway

This is an optional step that some users may want to perform if they want all traffic from the firewall to cross the VPN, not only LAN client traffic.

Policy routing is the most flexible way to direct traffic over this type of connection, but it does not influence traffic from the firewall itself. To send traffic from the firewall across the VPN to Internet destinations, the VPN must be set as the default gateway.

To avoid a chicken-and-egg problem, a manual static route is required for the VPN provider peer endpoint address:

- Navigate to **System > Routing**, **Static Routes** tab
- Click  **Add**
- Configure the routes as follows:

  Destination network The VPN provider peer endpoint IP address. For this example, `198.51.100.6`. Use a CIDR mask of `32` (or `128` if the peer endpoint is an IPv6 address.)

  Gateway *WANGW* so that traffic for this endpoint is routed over WAN

  Description `Route to VPN provider endpoint`

- Click **Save**

With the peer route in place, now set the default gateway:

- Navigate to **System > Routing**, **Gateways** tab

- Set **Default Gateway IPv4** to *WG_VPN_WGV4*, or a gateway group which includes that gateway, such as the previously created *Prefer_WireGuard*.
- Set **Default Gateway IPv6** in a similar manner if the VPN also carries IPv6 traffic.
- Click **Save**
- Click **Apply Changes**

At this point, all traffic that doesn't match entries in the routing table will be sent across the VPN.

34.59.9 DNS Configuration

DNS privacy is also important, and there are a few factors to consider. For this example, DNS requests will be sent to a DNS server at the VPN peer, but without TLS.

See also:

Options such as *DNS over TLS* are covered elsewhere, but can help as well.

First, set the VPN provider DNS server:

- Navigate to **System > General**
- Remove any DNS servers present in the list under **DNS Server Settings**
- Set a DNS Server entry as follows:

  Address The address of the DNS server at the peer, in this example, `10.6.0.1`.

  DNS Hostname If this server supports DNS over TLS, enter its hostname here. Otherwise, leave it blank.

  Gateway Select the VPN gateway, *WG_VPN_WGV4*.
- *Uncheck* **DNS Server Override** to prevent this firewall from using DNS servers from dynamic WANs.
- Set **DNS Resolution Behavior** based on the requirements of this environment:

> **Warning:** This can help prevent DNS requests from leaking to other servers not using the VPN, but it can cause a chicken-end-egg scenario where DNS requests will fail unless the VPN is working. Ensure that DNS is not required to establish the VPN.

  Use local DNS, fall back to remote DNS Servers Use this option when using the DNS Resolver in forwarding mode and when the DNS server does not need DNS over TLS. This is the best fit for this example.

  Use local DNS, ignore remote DNS Servers Use this option when using DNS over TLS with the DNS Resolver in forwarding mode. This ensures that no DNS query will be sent without TLS.

  Use remote DNS Servers, ignore local DNS Use this option if the firewall itself shouldn't use the DNS Resolver, but communicate directly with the DNS server without TLS.

Next, configure the DNS Resolver for Forwarding mode:

- Navigate to **Services > DNS Resolver**
- *Uncheck* **Enable DNSSEC Support**
- *Check* **Enable Forwarding Mode**
- Click **Save**
- Click **Apply Changes**

Warning: If there are any **Custom Options** in the DNS Resolver, it is possible that switching to forwarding mode will change the context of the options. Add `server:` to the beginning of the **Custom Options** box content, above any existing options. If the **Custom Options** box is empty, it can remain empty.

34.59.10 Finish Up

The configuration is now complete! Depending on which sections were followed, the firewall should be able to at least communicate with the remote peer, networks, and clients should be able to pass traffic through the VPN provider out to the Internet.

Make any final adjustments or additional configurations as needed.

See also:

- *WireGuard*
- *Routing*
- *Policy Routing Configuration*
- *Configuring DNS over TLS*
- *WireGuard Remote Access VPN Configuration Example*
- *WireGuard Site-to-Site VPN Configuration Example*

34.60 Accessing Port Forwards from Local Networks

By default, pfSense® software does not redirect internally connected devices to reach forwarded ports and 1:1 NAT on WAN interfaces. If a client is trying to reach a service on port 80 or 443 (or the port a web interface is using if it has been changed), the connection will hit the web interface and they will be presented with a certificate error if the GUI is running HTTPS, and a DNS rebinding error since it's an unrecognized hostname.

NAT Reflection employs techniques to redirect these connections if required. Split DNS is usually the better way if it is possible on a network because it allows for retaining of the original source IP and avoids unnecessarily looping internal traffic through the firewall. Both are explained here.

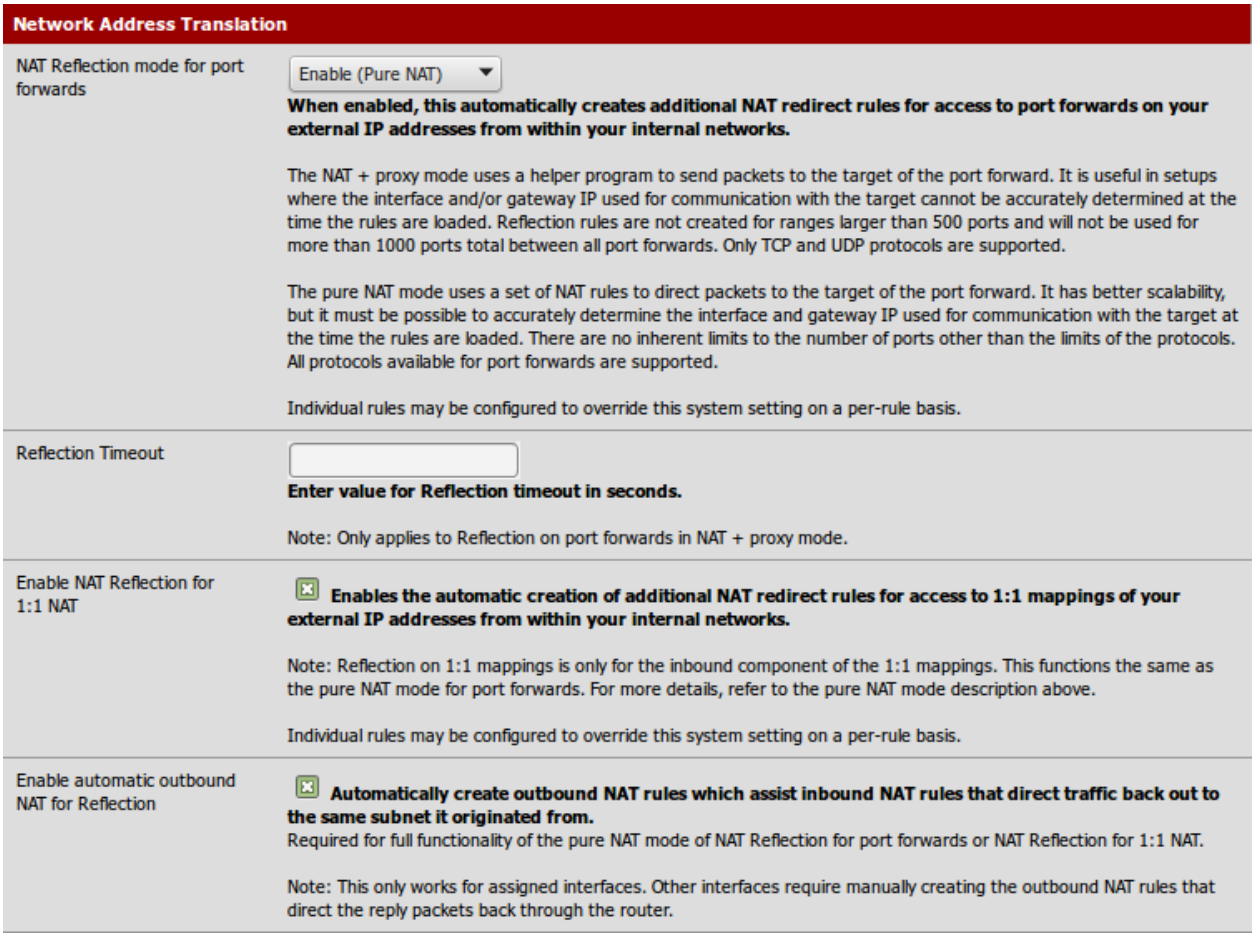
34.60.1 Method 1: NAT Reflection

In order to access ports forwarded on the WAN interface from internal networks, NAT reflection must be enabled.

In order to do this, navigate to **System > Advanced**, **Firewall/NAT** tab. On that page, select *Pure NAT* for **NAT Reflection mode for port forwards**, check **Enable NAT Reflection for 1:1 NAT**, and check **Enable automatic outbound NAT for Reflection**. Click **Save**.

Pure NAT mode for port forward reflections uses only pf NAT rules to accomplish reflection without any external daemons. It will work with TCP, UDP, and other protocols.

NAT+Proxy mode for port forward reflection sets up a proxy daemon and rules to receive and reflect only TCP connections. This method the only available means of reflection in earlier versions of pfSense software. It can work in certain rare circumstances where Pure NAT mode does not. This will only work with single port forwards or ranges of less than 500 ports. It does not work with UDP or other protocols.



34.60.2 Method 2: Split DNS

The more elegant solution to this problem involves using Split DNS. Basically this means that internal and external clients resolve hostnames differently.

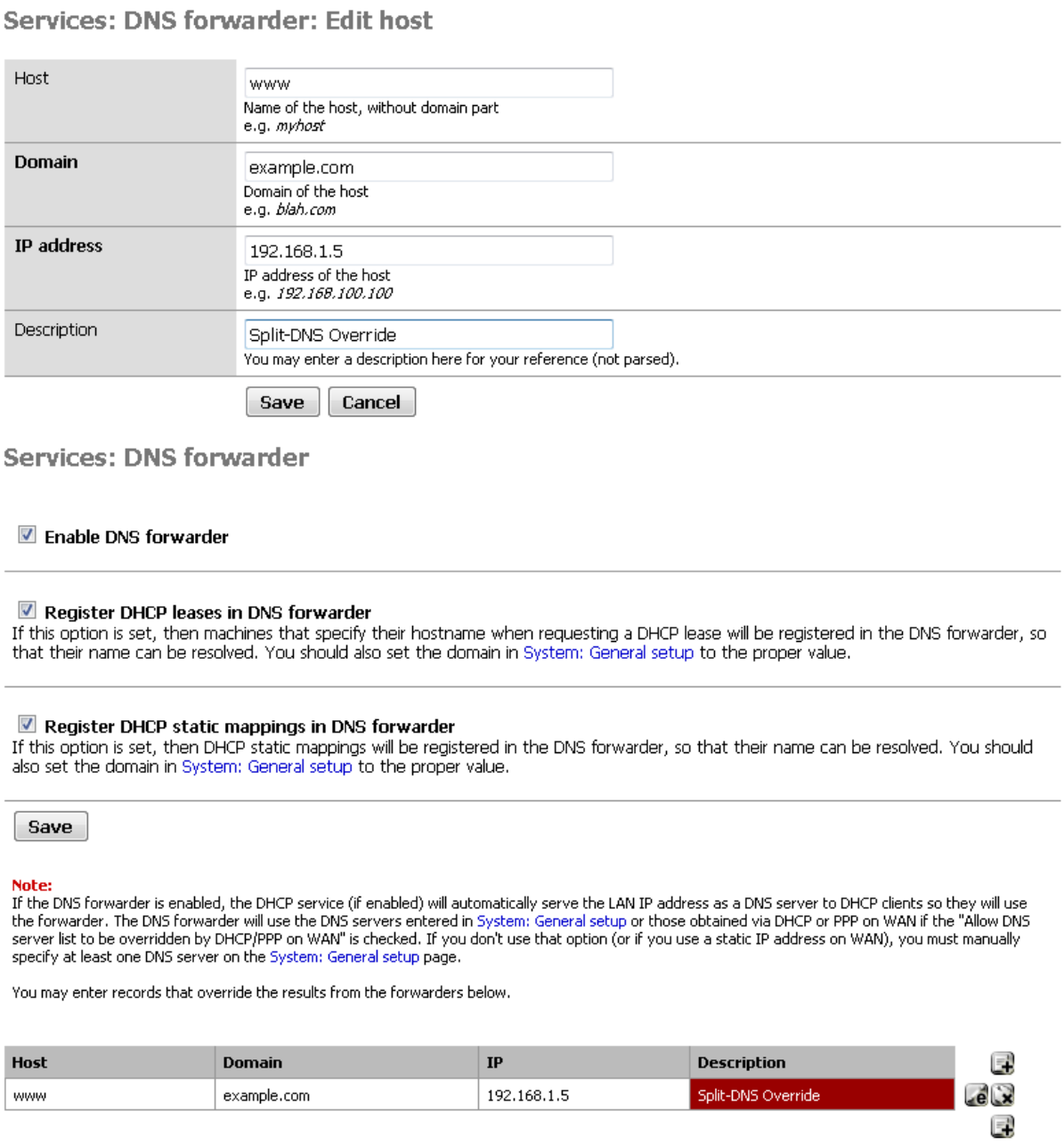
Internal clients would access resources by hostname, not IP, and clients on the local network would resolve that hostname to the LAN IP address of the actual server, and not the WAN IP as others outside the network would see.

In order for this to work using the DNS Forwarder or Resolver in pfSense software, clients will need to have the IP Address of the pfSense router as their primary DNS server.

Example:

- www.example.com resolves to public IP 1.2.3.4, which is the WAN IP
- Forward port 80 on 1.2.3.4 to port 80 on 192.168.1.5
- Override www.example.com using **Services > DNS Resolver** (or DNS Forwarder, if using it instead) and point www.example.com to 192.168.1.5
  - Another internal DNS mechanism could also be used to enact the override.

Screenshots that show the above in practice:


Services: DNS forwarder: Edit host
Host
www
Name of the host, without domain part
e.g. myhost
Domain
example.com
Domain of the host
e.g. blah.com
IP address
192.168.1.5
IP address of the host
e.g. 192.168.100.100
Description
Split-DNS Override
You may enter a description here for your reference (not parsed).
Save
Cancel
Services: DNS forwarder
Enable DNS forwarder
Register DHCP leases in DNS forwarder
If this option is set, then machines that specify their hostname when requesting a DHCP lease will be registered in the DNS forwarder, so that their name can be resolved. You should also set the domain in System: General setup to the proper value.
Register DHCP static mappings in DNS forwarder
If this option is set, then DHCP static mappings will be registered in the DNS forwarder, so that their name can be resolved. You should also set the domain in System: General setup to the proper value.
Save
Note:
If the DNS forwarder is enabled, the DHCP service (if enabled) will automatically serve the LAN IP address as a DNS server to DHCP clients so they will use the forwarder. The DNS forwarder will use the DNS servers entered in System: General setup or those obtained via DHCP or PPP on WAN if the "Allow DNS server list to be overridden by DHCP/PPP on WAN" is checked. If you don't use that option (or if you use a static IP address on WAN), you must manually specify at least one DNS server on the System: General setup page.
You may enter records that override the results from the forwarders below.
Host
Domain
IP
Description
www
example.com
192.168.1.5
Split-DNS Override

34.61 Authenticating from Active Directory using RADIUS/NPS

Windows 2008 and later can be configured as a RADIUS server using Microsoft's Network Policy Server (NPS). This allows authentication for OpenVPN, Captive Portal, the PPPoE server, or even the pfSense® GUI itself using Windows Server local user accounts or Active Directory.

34.61.1 Choosing a server for NPS

NPS requires a minimal amount of resources and is suitable for addition to an existing Windows Server in most environments. Microsoft recommends installing it on an Active Directory domain controller to improve performance in environments where NPS is authenticating against Active Directory. It can also be installed on a member server, which may be desirable in some environments to reduce the attack footprint of domain controllers. Each network-accessible service provides another potential avenue for compromising a server. NPS does have a solid security record, especially compared to other services that must be running on domain controllers for Active Directory to function, so this isn't much of a concern in most network environments. Most environments install NPS on one of their domain controllers. Microsoft recommends running it on each domain controller in the forest and using NPS proxies to share the load for a busy environment.

34.61.2 Installing NPS

On Windows Server 2008:

- Navigate to **Server Manager**
- Click **Roles** on the left and expand it
- Click **Add Roles** on the right
- Click **Next** to skip the intro screen

On Server 2012:

- Open the System Manager Dashboard
- Click **Add Roles and Features**
- Click past Role-based or feature-based installation
- Click **Next** once more
- Select the server from the list
- Click **Next** again

On either server version, the remaining steps are similar:

- Check **Network Policy and Access Services** on the list of roles
- Click **Add Features** if it appears
- Click **Next** on each screen until the end of the wizard
- Click **Finish** or **Install**, depending on the windows server version

34.61.3 Configuring NPS

To configure NPS, bring up the Server Manager and either Network Policy and Access Services (2008) or NAP (2012) should be present.

A RADIUS client will be added for pfSense first, then remote access policies will be configured.

Adding a RADIUS Client

Open the NPS configuration:

On Server 2008:

- Open the **Server Manager** tree
- Expand the view under it until **RADIUS Clients and Server** is visible
- Click **RADIUS Clients**

On Server 2012:

- Open the **Server Manager** dashboard
- Click **NAP**
- Right click on the server in the server list
- Click **Network Policy Server**
- Expand **RADIUS Clients and Server**
- Click **RADIUS Clients**

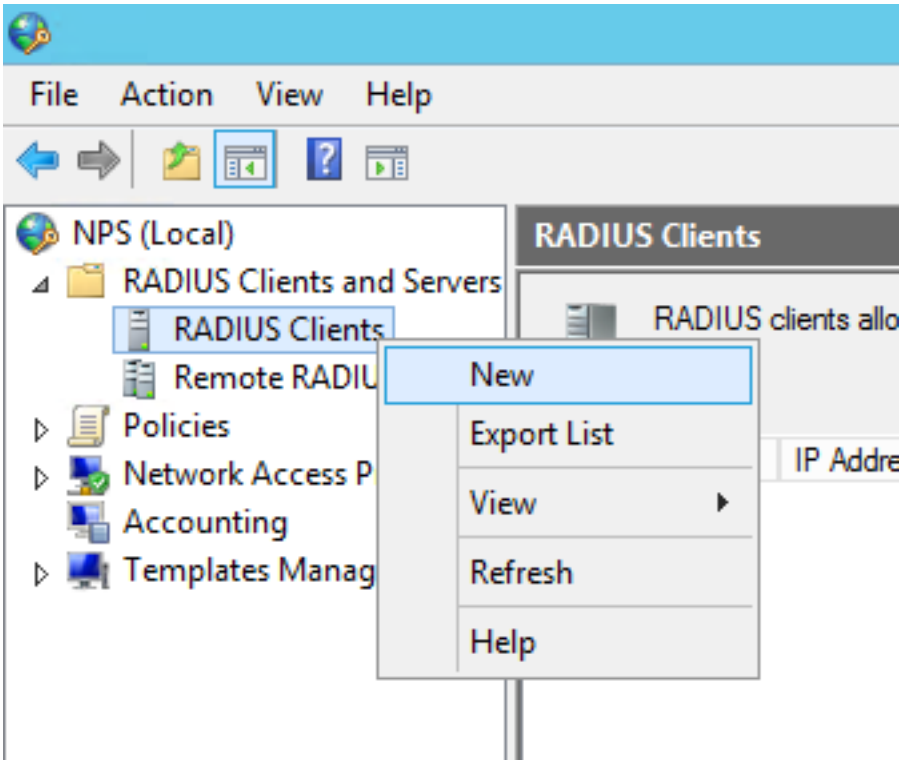


Fig. 82: Add New RADIUS Client

Add the new RADIUS client:

- Right click on **RADIUS Clients**
- Click **New**, as shown in Figure *Add New RADIUS Client*
- Enter a **Friendly name** for the firewall, as shown in Figure *Add New RADIUS Client Address*. This can be the hostname or FQDN.
- Enter the **Address (IP or DNS)** for the firewall, which must be the IP address from which pfSense will initiate RADIUS requests, or a FQDN that will resolve to that IP address.

Note: This is the IP address of the firewall interface closest to the RADIUS server. If the RADIUS server is reachable via the firewall LAN interface, this will be the LAN IP address. In deployments where pfSense is not the perimeter firewall, and the WAN interface resides on the internal network where the RADIUS server resides, the WAN IP address is what must be entered.

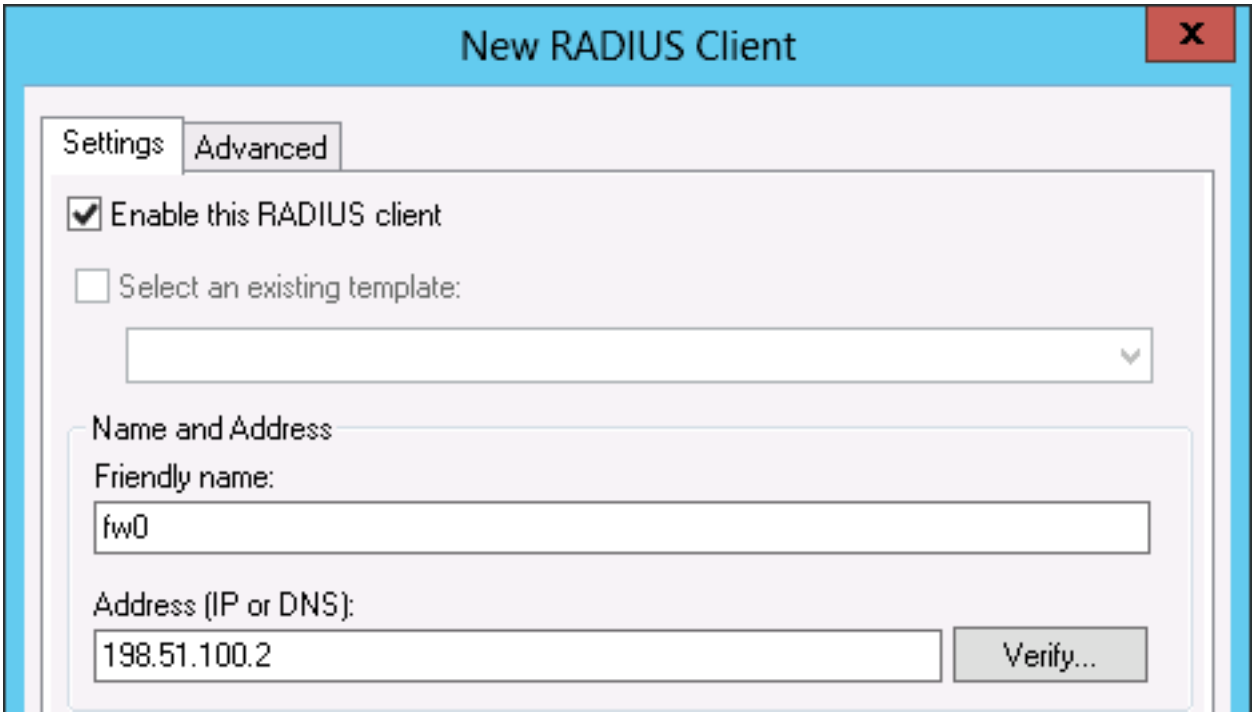


Fig. 83: Add New RADIUS Client Address

- Enter a **Shared secret**, as shown in Figure *Add New RADIUS Client Shared Secret*. This shared secret is used by pfSense to authenticate itself when making RADIUS access requests. Windows can automatically create one by clicking **Generate**.
- Click OK.

The NPS configuration is now complete. The RADIUS Client is visible as in Figure *Listing of the RADIUS Client*.

Refer to other sections in this documentation describing the service to be used with RADIUS for more guidance on how to utilize the service. RADIUS can be used in the User Manager (*User Management and Authentication*) which also enables RADIUS for IPsec and OpenVPN, for Captive Portal (*Portal Configuration Using RADIUS Authentication*), and the PPPoE server (*PPPoE Server*), among other places.

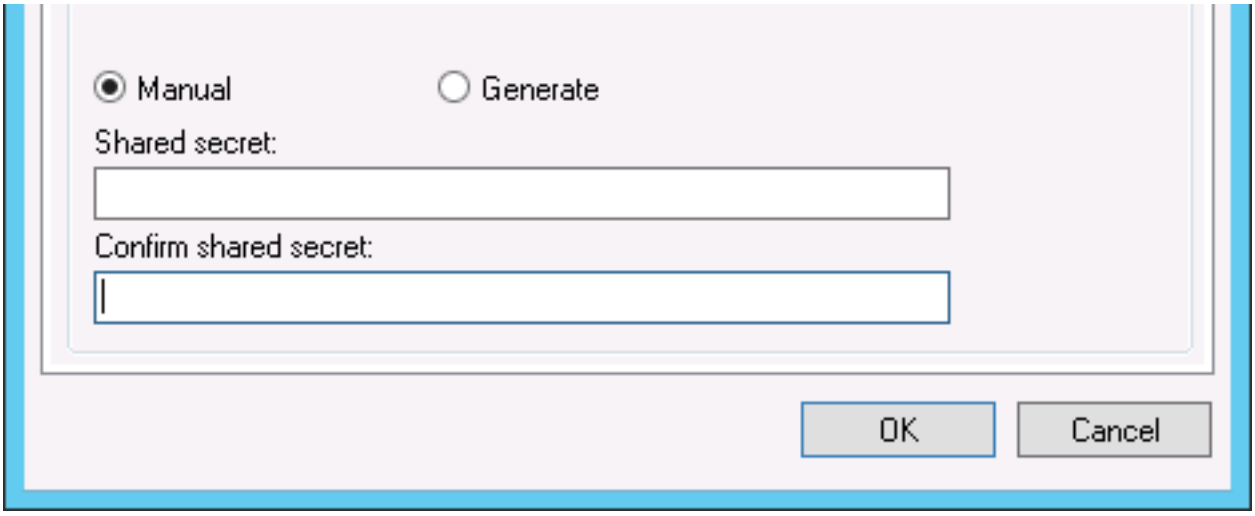


Fig. 84: Add New RADIUS Client Shared Secret

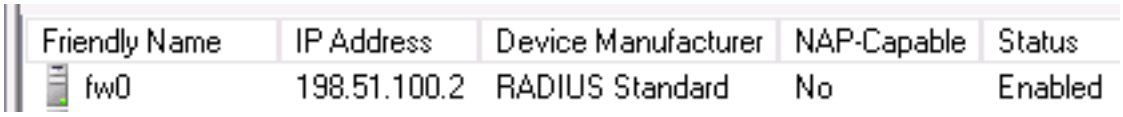


Fig. 85: Listing of the RADIUS Client

Configuring Users and Network Policies

Whether a user can authenticate via RADIUS is controlled through **Network Policies**. Using Network Policies, an administrator can place a user in a specific Active Directory group to allow VPN access, and also offer more advanced capabilities such as time of day restrictions.

More information on remote access policies can be found in Microsoft's documentation at http://technet.microsoft.com/en-us/library/cc785236%28WS.10%29.aspx.

Adding a Network Policy

- Open the NPS configuration window
- Expand **NPS (Local)**, **Policies**, then **Network Policies**
- Right click on **Network Policies**
- Click **New**
- Enter `Allow from pfSense` in the **Policy name**
- Leave the **Type of network access server** set to *Unspecified*
- Click **Next**
- Click **Add** in the Specify Conditions window
- Select **Windows Groups**
- Click **Add**
- Enter or select the name of the user group which contains VPN users, e.g. `VPNUsers`

- Click **OK**
- Click **Next**
- Choose **Access granted**
- Click **Next**
- Select additional **Authentication Methods** as needed for features on pfSense:
  - Leave existing authentication methods selected
  - Select **Microsoft: Secured Password (EAP-MSCHAP v2)** if this policy will be used for IPsec IKEv2 EAP-RADIUS authentication
  - Select **Encrypted Authentication (CHAP)**
  - Select **Unencrypted Authentication (PAP, SPAP)**

  leaving any other methods selected that were already enabled.
- Click **Next**
- Click **Decline** if a prompt to view a help topic is presented by the wizard
- Configure any additional access restraints, if necessary
- Click **Next** on the remaining screens until the final screen is reached
- Click **Finish**

Editing an Existing Network Policy

Existing policies can be altered to change their constraints or other properties. For example, to edit an older policy to enable it for use by IPsec for IKEv2 EAP-RADIUS:

- Open the NPS configuration window
- Expand **NPS (Local)**, **Policies**, then **Network Policies**
- Edit the policy currently in use
- Click the **Constraints** tab
- Click **Authentication Methods**
- Click **Add**
- Select **Microsoft: Secured Password (EAP-MSCHAP v2)**
- Click **OK**
- Click **Apply** to restart NPS
- Click **OK**

Troubleshooting NPS

If authentication fails, this section describes the most common problems users encounter with NPS.

Verify port

First ensure the default port **1812** is being used by NPS. If the NPS server was previously installed, it may have been configured with non-standard ports.

- Open the NPS configuration window
- Right click on **NPS (Local)** at the top left of the console
- Click **Properties**
- Click the **Ports** tab
- Verify the **Authentication** port configuration. Specify multiple ports by separating them with a comma. (as shown in Figure *NPS Ports*). Port `1812` **must** be one of the ports configured for Authentication.
- Verify the **Accounting** ports if necessary. If RADIUS accounting is required, port `1813` must be one of the ports specified in this box.

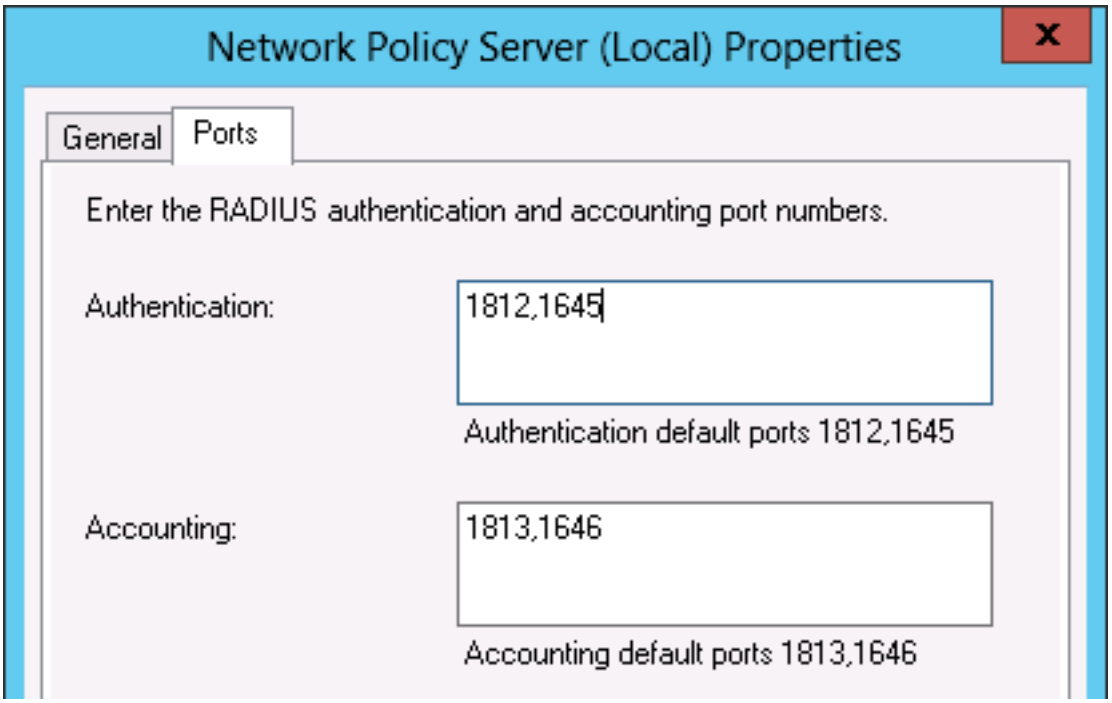


Fig. 86: NPS Ports

Check Event Viewer

When a RADIUS authentication attempt is answered by the server, NPS logs to the System log in Event Viewer with the result of the authentication request. If access is denied, the reason it was denied is logged.

In the Description field of the event properties, the Reason line tells why authentication failed. The common two failures are: bad username and password, when a user enters incorrect credentials; and "remote access permission for the user account was denied" when the user account is set to Deny access or the network policies configured in NPS do not allow access for that user. If NPS is logging that authentication was successful, but the client is receiving a bad username or password message, the RADIUS secret configured in NPS and pfSense does not match.

The NPS logs in Event Viewer may be easily found under Custom Views, then Server Roles, and finally Network Policy and Access Services.

34.62 Allowing Remote Access to the GUI

Several ways exist to remotely administer a firewall running pfSense® software that come with varying levels of recommendation. They all work, but their use may vary for any number of reasons (Client restrictions, corporate policies, etc.)

34.62.1 Use a VPN

The safest way to accomplish the task is to setup a VPN that will allow access to the firewall and the network it protects. There are several VPN options available in pfSense software, such as

- *IPsec*
- *OpenVPN*
- *SSH tunneling*

Once a VPN is in place, reach the GUI safely using a local address on the firewall, such as the LAN IP address. The exact details vary depending on the VPN configuration.

34.62.2 Restricted Firewall Access

If the webGUI port must be accessible to the Internet, restrict it by IP address/range as much as possible. Ideally, if there is a static IP address at the location to manage from, allow traffic from that IP address or subnet and nowhere else. Aliases also help, and they can include fully qualified domain names as well. If the remote management clients have a dynamic DNS address, add it to a management alias.

34.62.3 Use HTTPS

The best practice is to always use HTTPS to encrypt access to the GUI port. Modern browsers may complain about the certificate, but an exception can usually be stored so it will only complain the first time.

To disable (or re-enable) HTTPS for the GUI, navigate to **System > Advanced**, under the **Admin Access** tab, using the **Protocol** option in the webConfigurator section. See *Admin Access Tab* for details.

34.62.4 Move the GUI to an Alternate Port

Moving the GUI to a non-standard, random port is also beneficial. This does not improve the actual security of the GUI itself, but can potentially reduce the number of brute force attempts. The GUI can still be found by scanners unless the port is properly filtered.

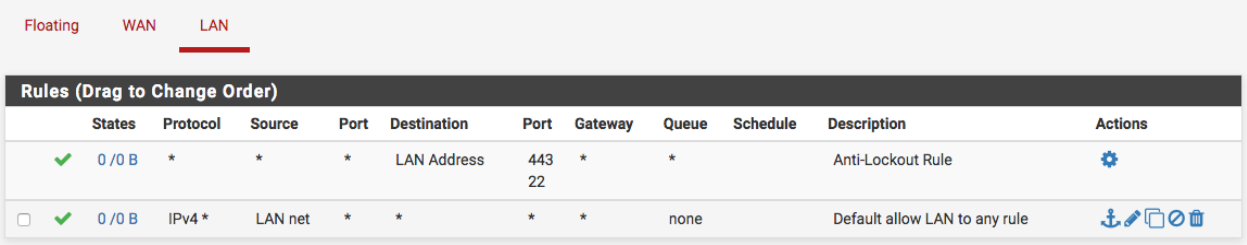
The port for the GUI can be changed under **System > Advanced**, **Admin Access** tab, using the *TCP Port* option in the **webConfigurator** section. Avoid common ports like `443`, `31337`, `8080`, `8888`, etc.

34.62.5 Strict Management

To enhance the security of a network, in many environments access to the firewall GUI is limited by firewall rules. Restricting access to the management interface is the best practice , for reasons as to why, see the blog post Securely Managing Web-administered Devices.

The default configuration of pfSense software allows management access from any machine on the LAN and denies it to anything outside of the local network. There is also an anti-lockout rule enabled by default that prevents firewall rules from being configured in a way that will lock the user out of the web interface.

To restrict management access first ensure the LAN rules allow access to the port used for the GUI. This depicts the default LAN rule, which allows access to the web interface.

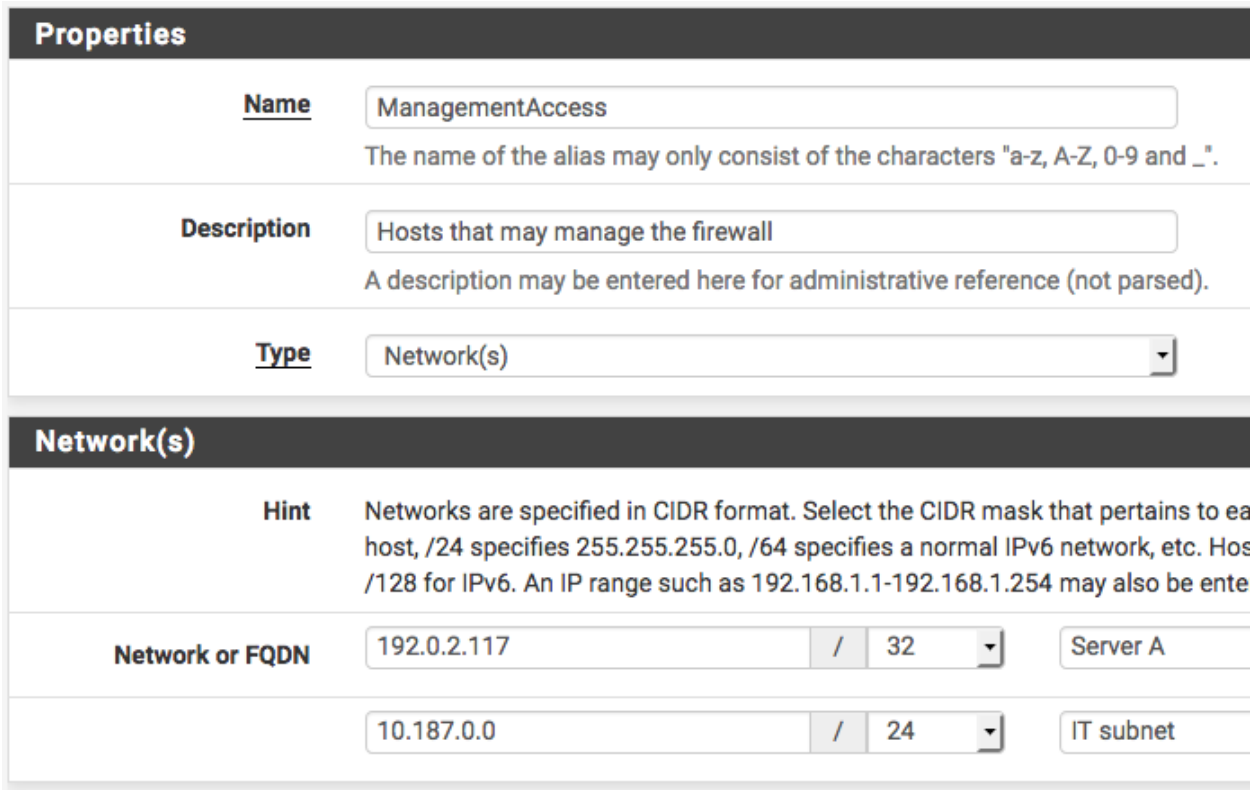


If a restrictive ruleset is in place on the LAN, make sure it permits access to the web interface before continuing.

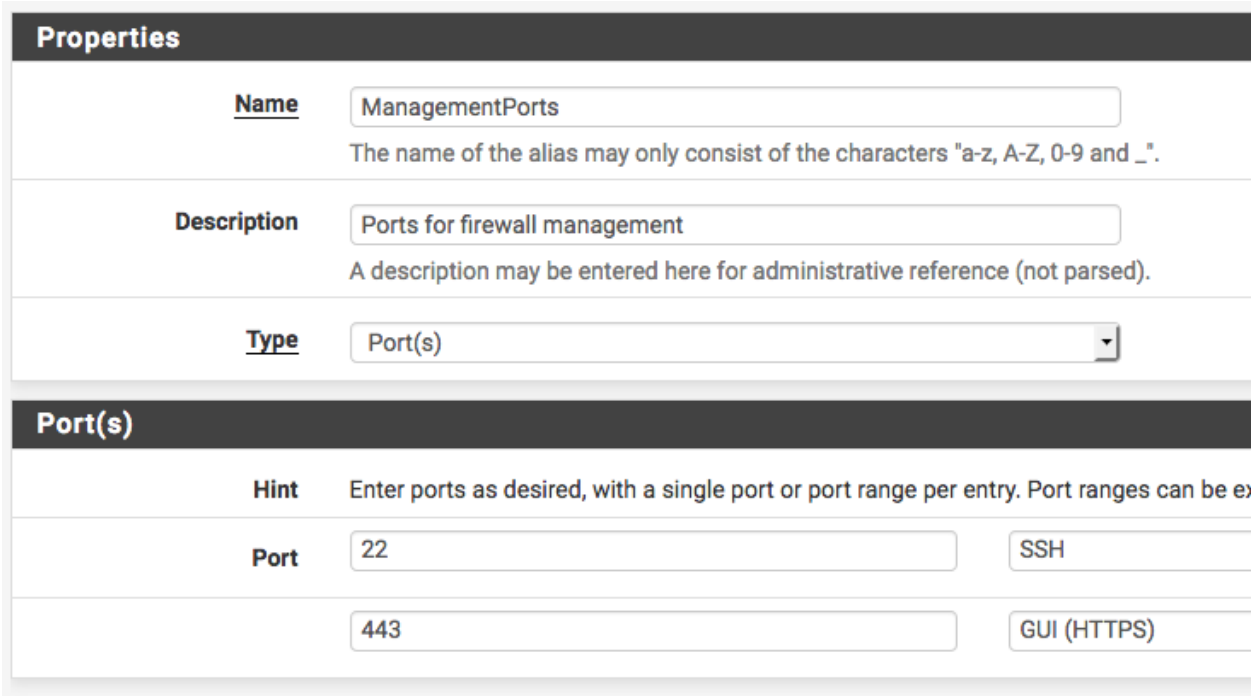
Now disable the anti-lockout rule. Navigate to **System > Advanced**, **Admin Access** tab and check **Disable webConfigurator anti-lockout rule**. Click **Save** and the rule will be removed.

Using a network alias for management access is another useful best practice. If both web and SSH administration are used, add an alias for those ports. The following are examples:

1. Example alias for networks allowed to access management interface

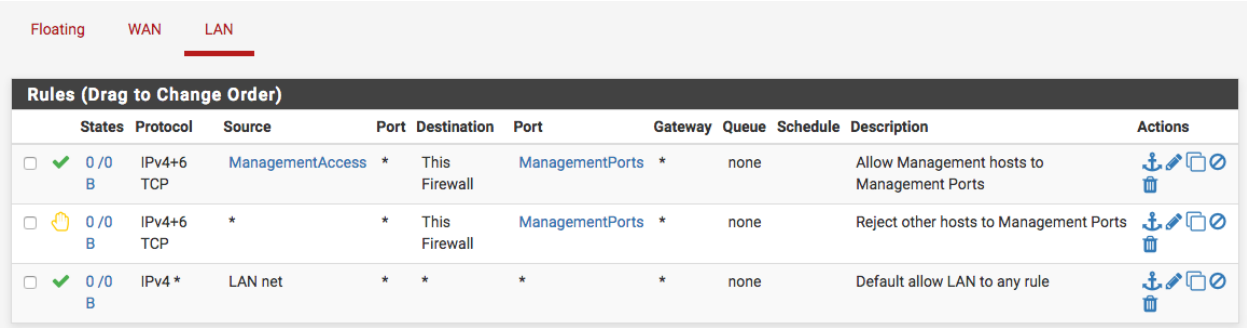


2. Example alias for ports allowed to access management interface



Now add a firewall rule allowing the sources defined in the management alias to the destination of the firewall, with the port used or alias created for those using multiple ports. **Make sure this rule is first in the list**. Then add a rule based on that rule (click  next to the rule), changing action to *block* or *reject* (reject is preferred on internal networks),

source to *any*, and destination the same. When finished the ruleset should look like the following.



Click **Apply Changes** and the management interface is now restricted to only the defined hosts.

34.62.6 I Don't Care About Security, How Do I Open Access To The GUI?

To open the firewall GUI, create a firewall rule to allow remote firewall administration.

Note: Do not create a port forward or other NAT configuration.

Firewall > Rules, WAN Tab

- **Action**: *pass*
- **Interface**: *WAN*
- **Protocol**: *TCP*
- **Source**: The IP address or subnet of the client, an alias containing management hosts/networks, or (as a last resort only) *Any*
- **Destination**: *WAN Address*
- **Destination port range**: *HTTPS* (Or the custom port)
- **Description**: *Allow remote management from anywhere (Dangerous!)*

34.63 Preventing RFC1918 Traffic from Exiting a WAN Interface

RFC1918 addresses are blocks of network IP addresses reserved for private use that are commonly used behind firewalls to allow a single public IP address to be shared with multiple devices using NAT. The default pfSense® installation assigns the 192.168.1.0/24 address space to the LAN interface, but RFC1918 also defines other CIDR ranges for private use:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

As a general rule, it is good practice to prevent network traffic intended for RFC1918 subnets from leaving the firewall via the WAN interface. This avoids unnecessary traffic on the WAN link, and also provides a small security benefit by keeping information about the LAN network behind the firewall.

An example where this rule might be helpful is if a machine on the local LAN (e.g. 192.168.1.5) is configured to access private LAN addresses that are routed across a VPN tunnel (e.g. 192.168.100.0/24). If the VPN link were

to go down, pfSense software would no longer have an active route for 192.168.100.0/24, and a packet intended for 192.168.100.0/24 will be routed out the WAN interface using the default route. This could potentially provide information about the private LAN to someone with access to the ISP's WAN network. A malicious user could even set up an imposter machine on the WAN network with a 192.168.100.0/24 address and pretend to be a machine on the inactive VPN link.

While the chance of this being a problem is small, the probability of unintentional RFC1918 traffic routing through the WAN interface will increase for installations with more complex LAN topologies, a large number of users (typos, etc), or routes that may frequently change (VPN, etc). In these scenarios, it may be beneficial to add a firewall rule preventing RFC1918 traffic from being routed out of the WAN interface.

34.63.1 Scenarios where RFC1918 addresses should NOT be blocked on the WAN interface

In its default configuration, pfSense software is not configured to block RFC1918 addresses from being routed from the LAN subnet to the outside WAN, because there are two common scenarios where blocking this traffic is not desirable:

- **ISP assigns a RFC1918 address to end users** - Some ISPs assign private network addresses to their customers and perform their own NAT for customer traffic to the public internet. Verify this by looking at the WAN interface IP address on the pfSense dashboard. If the assigned address is from one of the private IP ranges listed above, RFC1918 traffic should NOT be blocked.
- **pfSense is "chained" behind another device like a consumer firewall or wifi router (double NAT)** - In this case, pfSense performs NAT for devices connected to the pfSense LAN, and the WAN interface forwards traffic to the upstream device, where it undergoes a second NAT before entering the public internet. This is verified using the same steps as above - if the WAN IP address is from the RFC1918 range, do NOT block this traffic from exiting the WAN

This is an example of an RFC1918 address assigned to the pfSense WAN:



Warning: If either of these scenarios apply to the pfSense installation, do NOT add additional RFC1918 traffic blocking to the WAN interface as this may block LAN users from accessing the WAN.

34.63.2 Steps to block RFC1918 traffic from leaving the WAN interface

For installations where the above scenarios do not apply, an additional firewall rule can be put in place to prevent RFC1918 traffic from leaking out of the WAN interface. This provides a small increase in security and privacy by preventing information about the local LAN from being routed further upstream to the ISP.

To add a block rule for RFC1918 traffic, navigate to **Firewall > Aliases**:

- Create an alias for the RFC1918 network ranges. Call it *private_networks* and include the following ranges:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

  - (optionally include other non-public CIDR ranges like 169.254.0.0/16 and 127.0.0.0/8)

- Add a new floating firewall rule under **Firewall > Rules**, **Floating** tab
  - **Action** - *Reject*
  - **Quick** - *Checked*
  - **Interface** - *WAN* (optionally select multiple WAN interfaces or interface groups here, do NOT select the local LAN)
  - **Direction** - *out*
  - **TCP/IP version** - *IPv4*
  - **Protocol** - *any*
  - **Source** - *any*
  - **Destination** - *Single host or alias: private_networks*
  - **Log** - *optional*
- Save the changes and reload the firewall. Verify that local LAN and internet connectivity are still working.

34.63.3 Notes

Adding this rule to the pfSense firewall will block access to bridge devices like cable modems or upstream routers outside of the WAN interface. For example, many cable modems use an IP address of 192.168.100.1. This may or may not be desirable behavior for users. The RFC1918 firewall rule needs to be disabled if access from inside the LAN to a device like this is required.

On the edit interfaces screen (**Interfaces > WAN**, for example) there is an option to *Block private networks*. This is a rule blocking inbound traffic, not outbound like the rule described here. As long as the pfSense instance is not behind a WAN that uses private addressing, both rules are desirable and should be enabled.

34.64 Routing Public IP Addresses

This section covers the routing of public IP addresses where a public IP subnet is assigned to an internal interface on a single firewall deployment.

See also:

If a High Availability cluster is in use, see *High Availability Configuration Example without NAT*.

34.64.1 IP Assignments

At least two public IP subnets must be assigned by the ISP. One is for the WAN of the firewall, and one for the inside interface. This is commonly a /30 subnet for the WAN, with a second subnet assigned for the internal interface. This example will use a /30 on WAN as shown in Table *WAN IP Block* and a /29 public subnet on an internal OPT interface as shown in Table *Inside IP Block*.

Table 10: WAN IP Block

198.51.100.64/30	
IP Address	Assigned To
198.51.100.65	ISP router (pfSense® default gateway)
198.51.100.66	pfSense WAN interface IP address

Table 11: Inside IP Block

192.0.2.128/29	
IP Address	Assigned To
192.0.2.129	pfSense OPT interface
192.0.2.130	Internal hosts
192.0.2.131	
192.0.2.132	
192.0.2.133	
192.0.2.134	

34.64.2 Interface Configuration

First configure the WAN and OPT interfaces. The LAN interface can also be used for public IP addresses if desired. In this example, LAN is a private IP subnet and OPT1 is the public IP subnet.

Configure WAN

Add the IP address and gateway accordingly. Figure *WAN IP and Gateway Configuration* shows the WAN configured as shown in Table *WAN IP Block*.



Fig. 87: WAN IP and Gateway Configuration

Configure OPT1

Now enable OPT1, optionally change its name, and configure the IP address and subnet mask. Figure *Routing OPT1 Interface Configuration* shows OPT1 configured as shown in Table *Inside IP Block*.

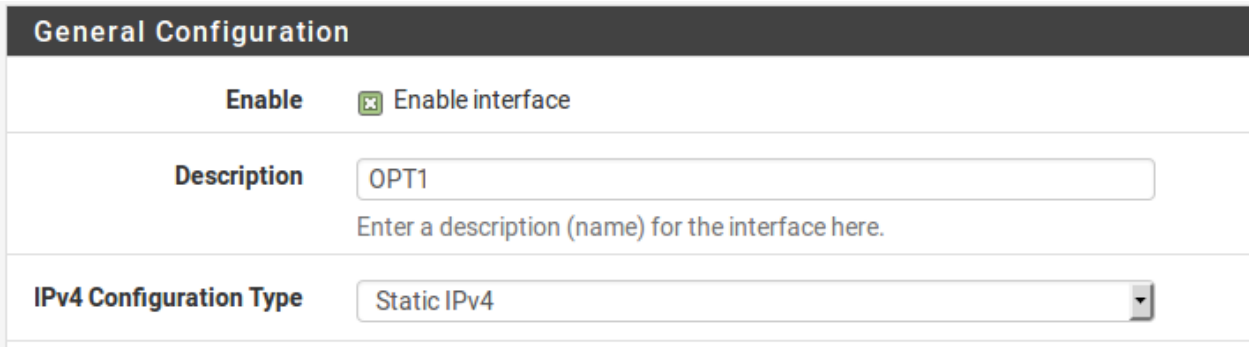


Fig. 88: Routing OPT1 Interface Configuration



Fig. 89: Routing OPT1 IP Address Configuration

34.64.3 NAT Configuration

The default of translating internal traffic to the WAN IP must be overridden when using public IP addresses on an internal interface.

- Browse to **Firewall > NAT**
- Click the Outbound tab
- Select Hybrid Outbound NAT rule generation
- Click **Save**
- Click  to add a new rule to the top of the list with the following settings:

  Do not NAT Checked, so that NAT will be disabled

  Interface *WAN*

  Protocol *Any*

  Source Network, enter the local public IP subnet, `192.0.2.128/29`

  Destination *Any*

- Click **Save**

This will override the default automatic rules which translate all traffic from local interfaces leaving the WAN interface to the WAN IP address. Traffic sourced from the OPT1 network `192.0.2.128/29` is not translated because of the manually added rule excluding it from NAT. This configuration maintains the automatic behavior for other internal interfaces, so that the advantages of automatic outbound NAT rules are not lost. This configuration is shown in Figure *Outbound NAT Configuration*.

If public IP addresses are used on **all** local interfaces, then set **Disable Outbound NAT** rather than using Hybrid mode.

34.64.4 Firewall Rule Configuration

The NAT and IP address configuration is now complete. Firewall rules will need to be added to permit outbound and inbound traffic. Figure *OPT1 Firewall Rules* shows a DMZ-like configuration, where all traffic destined for the LAN subnet is rejected, DNS and pings to the OPT1 interface IP address are permitted, and HTTP is allowed outbound.

To allow traffic from the Internet to the public IP addresses on an internal interface, add rules on the WAN using the public IP addresses as the **Destination**. Figure *WAN Firewall Rules* shows a rule that allows HTTP to 192.0.2.130, one of the public IP addresses on the internal interface as shown in Table *Inside IP Block*.

After configuring the firewall rules as desired, the setup is complete.

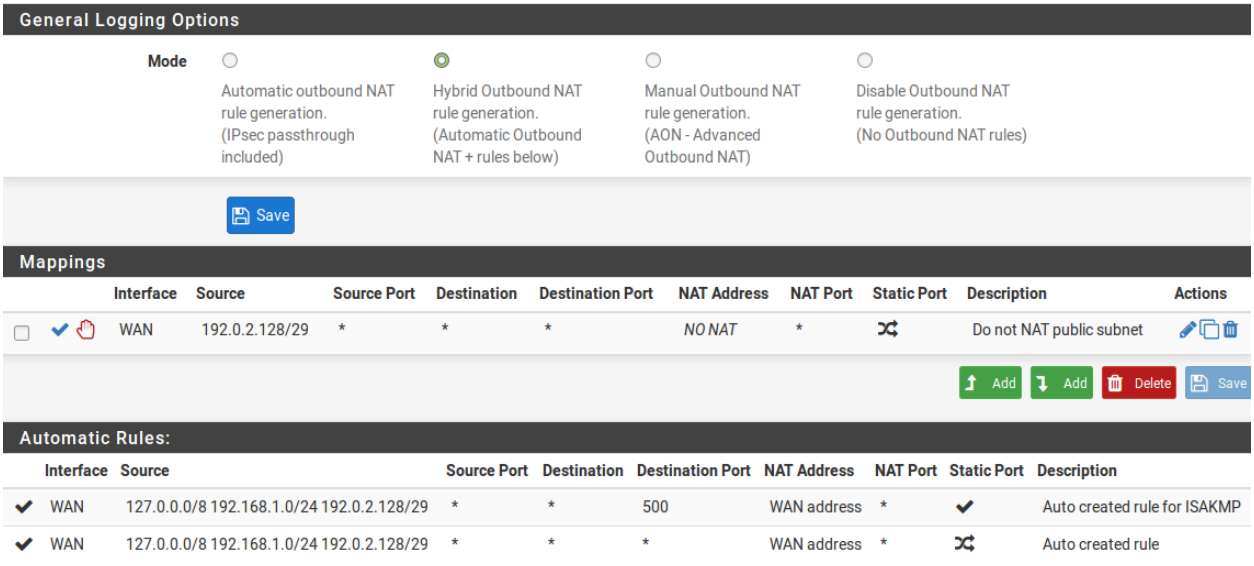


Fig. 90: Outbound NAT Configuration

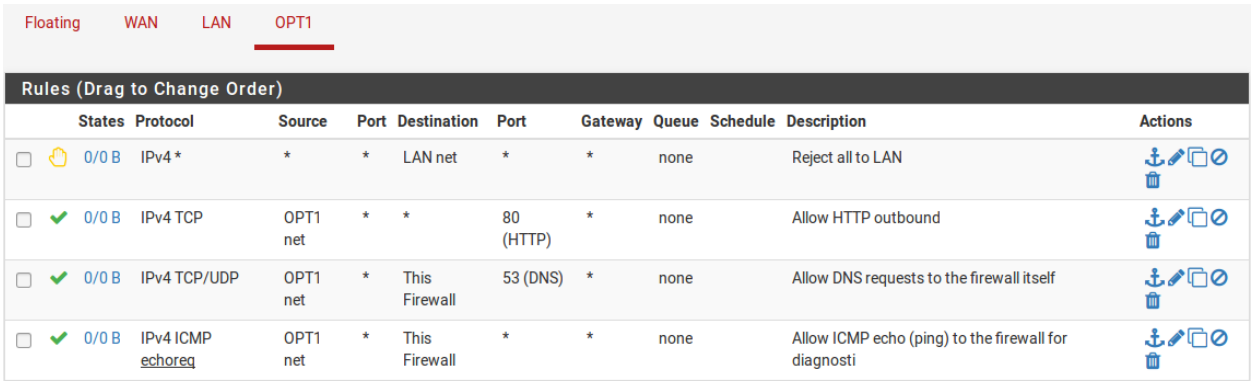


Fig. 91: OPT1 Firewall Rules

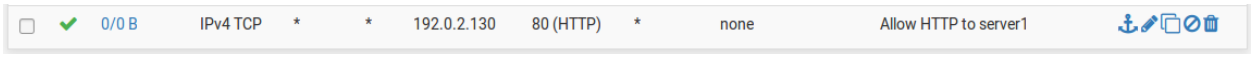


Fig. 92: WAN Firewall Rules

Note: Traffic will flow from LAN to this public subnet by default without NAT. If this behavior is not desired, adjust the LAN firewall and NAT rules accordingly. Additionally, policy routing may need to be bypassed to allow from LAN to this interface.

34.65 Accessing the Firewall Filesystem with SCP

Files may be transferred to and from the firewall with `scp`, which is part of the functionality that comes with having ssh access enabled.

See also:

- *Secure Shell (SSH)*
- *Granting Users Access to SSH*

To connect to the firewall with SCP for file transfers, use the `root` account with the same credentials as `admin`, or a user account with sufficient privileges.

Note: In most recent builds, the `admin` account can also be used for `scp`, but using `root` is still the best practice.

Users with shell access may transfer files, as well as users with the *User - System - Copy files* privilege.

Note: Users other than `root` can only transfer or write files for which their account has permission to read or modify.

Any SCP/SFTP-compatible program may be used to transfer files. Popular choices include `scp`, FileZilla, and WinSCP.

An example: Logged into a FreeBSD machine, copy a file from `/tmp` on a remote firewall to the current local working directory:

```
myuser@somebox:~/$ scp root@192.168.1.1:/tmp/lan-traffic.rrd-4h.png .
```

34.65.1 Troubleshooting

If `scp` is failing with the error `Illegal option -t` or printing a usage message, check that the connection is using the `root` account, as mentioned above.

34.66 Granting Users Access to SSH

This recipe explains how to enable *Secure Shell (SSH)* access to the firewall.

SSH is typically used for debugging and troubleshooting, but has many other useful purposes.

Note: The SSH daemon is not required by the firewall for operation, so it is disabled by default.

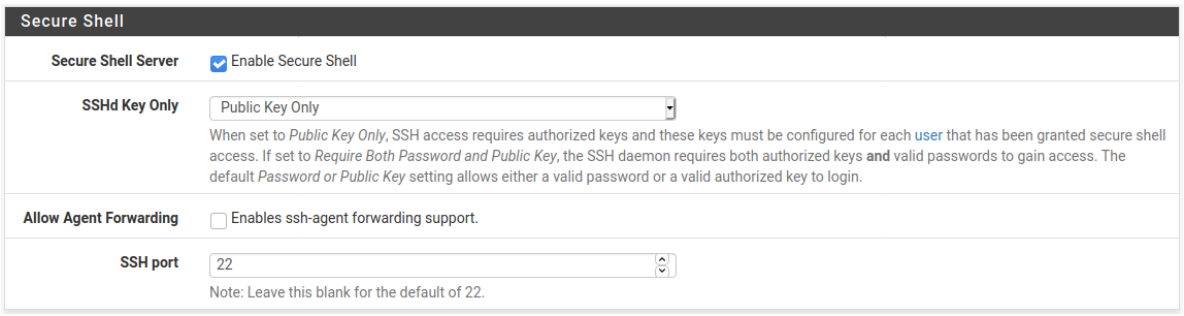
See also:

Secure Shell (SSH)

34.66.1 Enable SSH via GUI

This example enables SSH access using only public key authentication, which is more secure than allowing access by password alone.

- Navigate to **System > Advanced**, **Admin Access** tab
- Check **Enable Secure Shell**
- Set **SSHd Key Only** to *Public Key Only* to allow only key-based SSH authentication



- Enter a port number in **SSH Port** if the SSH daemon should listen on a non-default port

  Leave the field blank for the daemon to use port `22`

- Click **Save**

34.66.2 SSH Keys

When the SSH daemon is set for key-based authentication, it uses the keys defined on user accounts. Add keys to individual user accounts under **System > User Manager**. The `admin` user and `root` user share keys.

> **Warning:** Do not attempt to manage keys from the shell directly.

See also:

Manage Local Users

34.66.3 Enable SSH via Console

Connect to the console (VGA or Serial) and use option `14` to enable or disable SSH.

To change the port number or key authentication options, use the GUI as directed above.

34.66.4 SSH Daemon Security

With a default ruleset, SSH may only be accessed by clients on the LAN. If SSH access must be allowed for clients the WAN, the best practice is to restrict access to Key-based authentication to avoid issues with brute force attacks. Moving the daemon to an alternate port is also a good practice, but moving the port alone is not sufficient protection.

The firewall will automatically block users who attempt to authenticate unsuccessfully. This behavior, and settings to control it, are described in *Login Protection*.

If password authentication is active, ensure that all user accounts with shell access have strong passwords that cannot be easily guessed.

See also:

See *Best Practices for SSH* for more on SSH security.

34.66.5 User Access

By default only `admin` and `root` have SSH access. Additional users with limited access may be granted the *User - System - Shell account access* privilege to login via SSH.

Note: Additional users do not have full root privileges in the shell, so the system does not display the console menu for those users. Many commands and other files are inaccessible as well. For a normal user to get much use from the shell, the *Sudo Package* can delegate additional privileges to run commands as *root* or other users.

34.66.6 SCP File Transfers

For information on using SCP file transfers via SSH, see *Accessing the Firewall Filesystem with SCP*.

34.67 Configuring Switches with VLANs

This section provides guidance on configuring a few varieties of switches for use with VLANs. This offers generic guidance that will apply to most if not all 802.1Q capable switches, then goes on to cover configuration on specific switches from Cisco, HP, Netgear, and Dell.

Note: This is the bare minimum configuration needed for VLANs to function, and it does not necessarily show the ideal secure switch configuration for any specific environment. An in depth discussion of switch security is outside the scope of this documentation.

34.67.1 Switch configuration overview

Generally three or four things must be configured on VLAN capable switches:

1. **Add/define the VLANs**

   Most switches have a means of defining a list of configured VLANs, and they must be added before they can be configured on any ports.

2. **Configure the trunk port**

   The port to which the pfSense® firewall will be connected must be configured as a trunk port, tagging all possible VLANs on the interface.

3. **Configure the access ports**

   Configure ports for internal hosts as access ports on the desired VLANs, with **untagged** VLANs.

4. **Configure the Port VLAN ID (PVID)**

   Some switches require configuring the PVID for access ports. This specifies which VLAN to use for the traffic entering that switch port. For some switches this is a one step process, by configuring the port as an access port on a particular VLAN, it automatically tags traffic coming in on that port. Other switches require this to be configured in one or two places. Check the switch documentation for details if it is not one detailed in this chapter.

34.67.2 Cisco IOS based switches

Configuring and using VLANs on Cisco switches with IOS is a fairly simple process, taking only a few commands to create and use VLANs, trunk ports, and assigning ports to VLANs. Many switches from other vendors behave similarly to IOS, and will use nearly the same if not identical syntax for configuration.

Create VLANs

VLANs can be created in a standalone fashion, or using VLAN Trunk Protocol (VTP). Using VTP may be more convenient, as it will automatically propagate the VLAN configuration to all switches on a VTP domain, though it also can create its own security problems and open up possibilities for inadvertently wiping out the VLAN configuration.

With VTP, to add another VLAN it only needs to be configured on a single switch, and then all other trunked switches in the group can assign ports to that VLAN. If VLANs are configured independently, they must be added to each switch by hand. Refer to Cisco's documentation on VTP to ensure a secure configuration use used, and that it is not prone to accidental destruction.

In a network with only a few switches where VLANs do not change frequently, VTP may be overkill and avoiding it will also avoid its potential downfalls.

Standalone VLANs

To create standalone VLANs:

```
sw# vlan database
sw(vlan)# vlan 10 name "DMZ Servers"
sw(vlan)# vlan 20 name "Phones"
sw(vlan)# exit
```

VTP VLANs

To setup a switch for VTP and VLANs, create a VTP database on the master switch and then create two VLANs:

```
sw# vlan database
sw(vlan)# vtp server
sw(vlan)# vtp domain example.com
sw(vlan)# vtp password SuperSecret
sw(vlan)# vlan 10 name "DMZ Servers"
sw(vlan)# vlan 20 name "Phones"
sw(vlan)# exit
```

Configure Trunk Port

For pfSense, a switch port not only has to be in trunk mode, but also must be using 802.1q tagging. This can be done like so:

```
sw# configure terminal
sw(config)# interface FastEthernet 0/24
sw(config-if)# switchport mode trunk
sw(config-if)# switchport trunk encapsulation dot1q
```

Note: On some newer Cisco IOS switches, the Cisco-proprietary ISL VLAN encapsulation method is deprecated and no longer supported. If a switch does not allow the `encapsulation dot1q` configuration option, it only supports 802.1Q and the encapsulation does not need to be specified.

Add Ports to the VLAN

To add ports to these VLANs, assign them as follows:

```
sw# configure terminal
sw(config)# interface FastEthernet 0/12
sw(config-if)# switchport mode access
sw(config-if)# switchport access vlan 10
```

34.67.3 Cisco CatOS based switches

Creating VLANs on CatOS is a little different, though the terminology is the same as using VLANs under IOS. Standalone VLANs and VTP are both possible to maintain the VLAN database:

```
# set vtp domain example mode server
# set vtp passwd SuperSecret
# set vlan 10 name dmz
# set vlan 20 name phones
```

Then configure a trunk port to automatically handle every VLAN:

```
# set trunk 5/24 on dot1q 1-4094
```

Then add ports to the VLAN:

```
# set vlan 10 5/1-8
# set vlan 20 5/9-15
```

34.67.4 HP ProCurve switches

HP ProCurve switches only support 802.1q trunking, so no configuration is needed for encapsulation. First, ssh or telnet into the switch and bring up the management menu.

Enable VLAN Support

First, VLAN support needs to be enabled on the switch if it is not already:

1. Choose **Switch configuration**
2. Choose **Advanced Features**
3. Choose **VLAN Menu...**
4. Choose **VLAN Support**
5. Set **Enable VLANs** to *Yes* if it is not already, and choose a number of VLANs. Each time this value is changed the switch must be restarted, so ensure it is large enough to support as many VLANs as necessary.
6. Restart the switch to apply the changes.

Create VLANs

Before the VLANs can be assigned to ports, The VLANs must be created. At the switch configuration menu:

1. Choose **Switch configuration**
2. Choose **Advanced Features**
3. Choose **VLAN Menu...**
4. Choose **VLAN Names**
5. Choose **Add**
6. Enter the **VLAN ID**, `10`
7. Enter the **name**, `DMZ`
8. Choose **Save**
9. Repeat the steps from **Add** to **Save** for any remaining VLANs

Assigning Trunk Ports to VLANs

Next, configure the trunk port for the firewall as well as any trunk ports going to other switches containing multiple VLANs.

1. Choose **Switch configuration**
2. Choose **VLAN Menu...**
3. Choose **VLAN Port Assignment**
4. Choose **Edit**

5. Find the port to assign
6. Press **space** on Default VLAN until it shows **No**
7. Move over to the column for each of the VLANs on this trunk port, and Press **space** until it shows **Tagged**. Every VLAN in use must be tagged on the trunk port.

Assigning Access Ports to VLANs

1. Choose **Switch configuration**
2. Choose **VLAN Menu…**
3. Choose **VLAN Port Assignment**
4. Choose **Edit**
5. Find the port to assign
6. Press **space** on **Default VLAN** until it shows **No**
7. Move over to the column for the VLAN to which this port will be assigned
8. Press **space** until it shows **Untagged**.

34.67.5 Netgear Managed Switches

This example is on a GS108Tv1, but other Netgear models are all very similar if not identical. There are also several other vendors including Zyxel who sell switches made by the same manufacturer, using the same web interface with a different logo. Log into the web interface of the switch to start.

Planning the VLAN configuration

Before configuring the switch, several items are required:

1. The number of VLANs to be configured
2. The IDs to use for the VLANs
3. How each switch port needs to be configured

For this example, an 8 port GS108Tv1 is used, and it will be configured as shown in Table *Netgear GS108T VLAN Configuration*.

Table 12: Netgear GS108T VLAN Configuration

Switch port	VLAN mode	VLAN assigned
1	trunk	*10* and *20*, tagged
2	access	*10* untagged
3	access	*10* untagged
4	access	*10* untagged
5	access	*20* untagged
6	access	*20* untagged
7	access	*20* untagged
8	access	*20* untagged

Enable 802.1Q VLANs

To configure the switch to use 802.1Q VLAN trunking:

- Navigate to the **System** menu on the left side of the page
- Click **VLAN Group Setting**, as indicated in Figure *VLAN Group Setting*.



Fig. 93: VLAN Group Setting

- Select IEEE 802.1Q VLAN (Figure *Enable 802.1Q VLANs*).

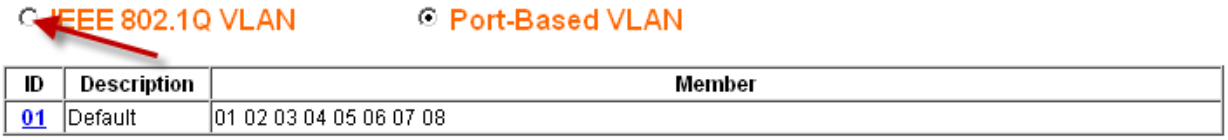


Fig. 94: Enable 802.1Q VLANs

- Click **OK** to confirm the switch to 802.1Q trunking, as shown in Figure *Confirm change to 802.1Q VLAN*.

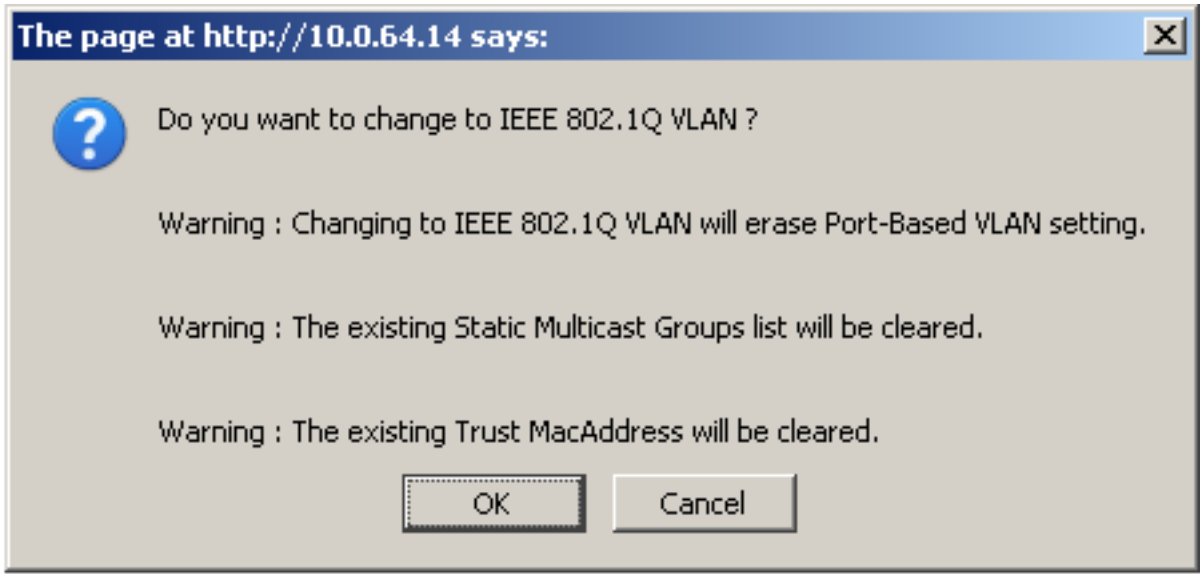


Fig. 95: Confirm change to 802.1Q VLAN

After clicking OK, the page will refresh with the 802.1Q VLAN configuration as shown in Figure *Default 802.1Q Configuration*.

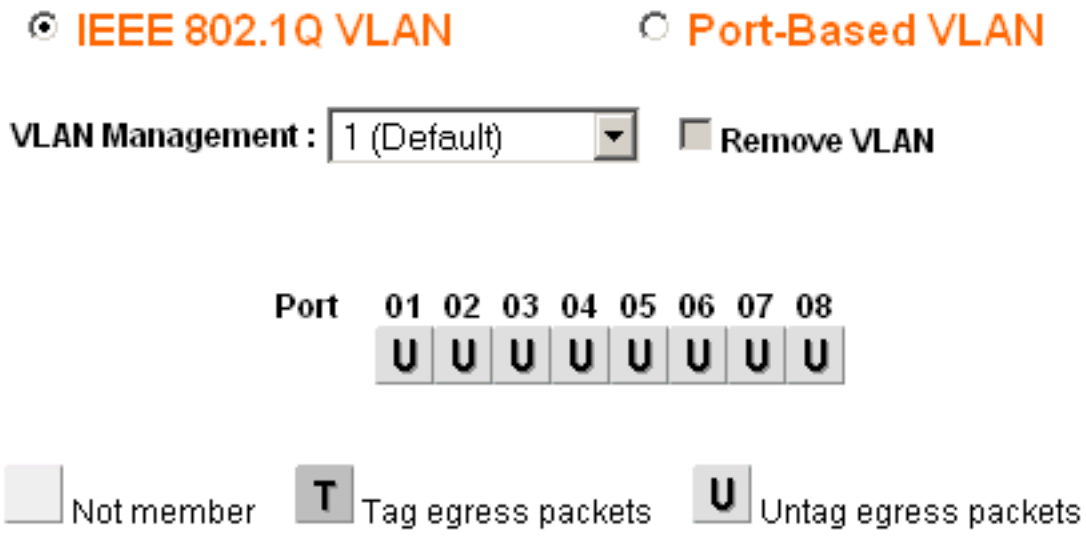


Fig. 96: Default 802.1Q Configuration

Add VLANs

For this example, two VLANs are added with IDs `10` and `20`. To add a VLAN:

- Click the **VLAN Management** drop down
- Click **Add New VLAN** as shown in Figure *Add New VLAN*.

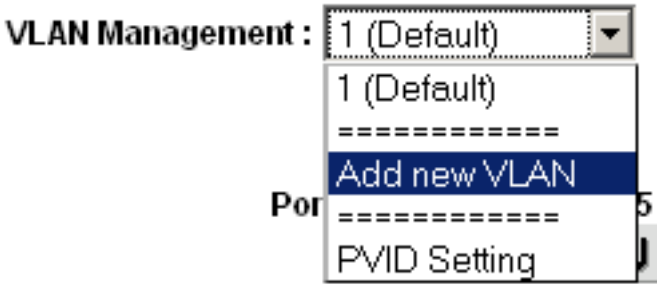


Fig. 97: Add New VLAN

- Enter the VLAN ID for this new VLAN, such as `10`
- Click **Apply**. The VLAN screen is now ready to configure VLAN *10* (Figure *Add VLAN 10*).
- Click **Add New VLAN** again as shown in Figure *Add New VLAN* to add VLAN *20* (Figure *Add VLAN 20*).

Add as many VLANs as needed, then continue to the next section.

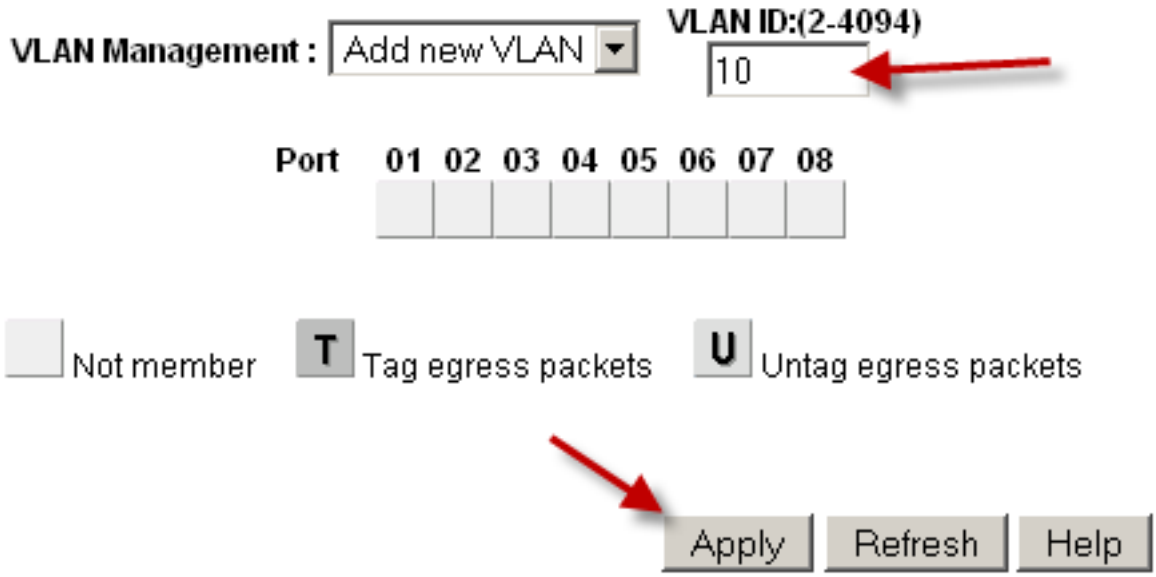


Fig. 98: Add VLAN 10

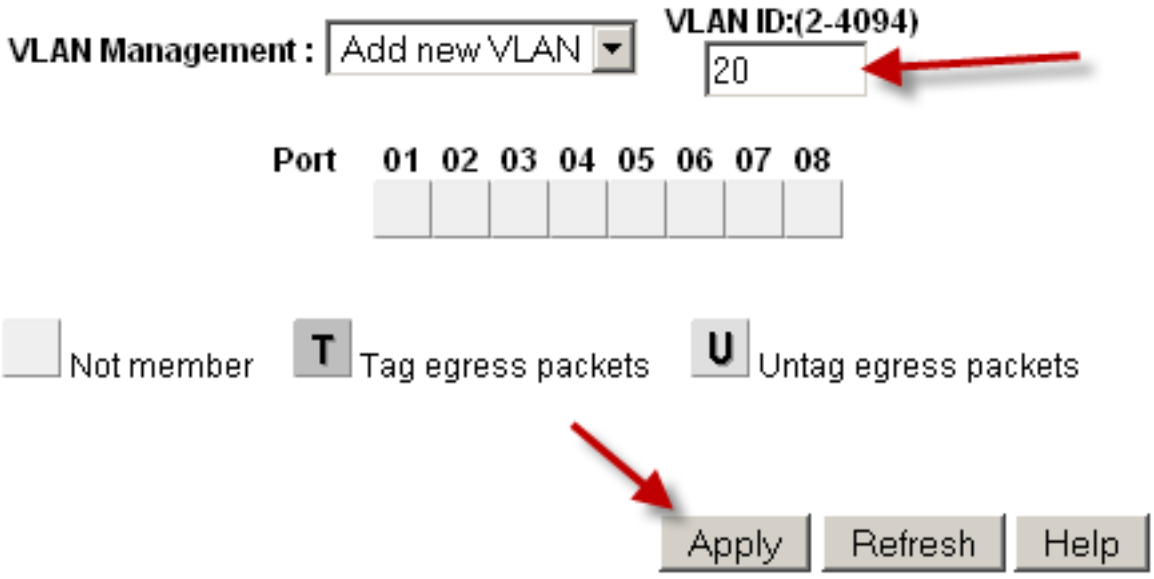


Fig. 99: Add VLAN 20

Configure VLAN tagging

When a VLAN is selected from the **VLAN Management** drop down, it shows how that VLAN is configured on each port:

- A **blank** box means the port is not a member of the selected VLAN.
- A box containing **T** means the VLAN is sent on that port with the 802.1Q tag.
- **U** indicates the port is a member of that VLAN and it leaves the port untagged.

The trunk port must have both VLANs added and tagged.

> **Warning:** Do not change the configuration of the port being used to access the web interface of the switch! This will lock the administrator out of the switch. The only means of recovery on the GS108Tv2 is using the **reset to factory defaults** button since it does not have a serial console. For the switches that have serial consoles, keep a null modem cable handy in case network connectivity with the switch is lost. Configuring the management VLAN is covered later in this section.

Click in the boxes beneath the port number as shown in Figure ref:*figure-toggle-vlan-membership* to toggle between the three VLAN options.

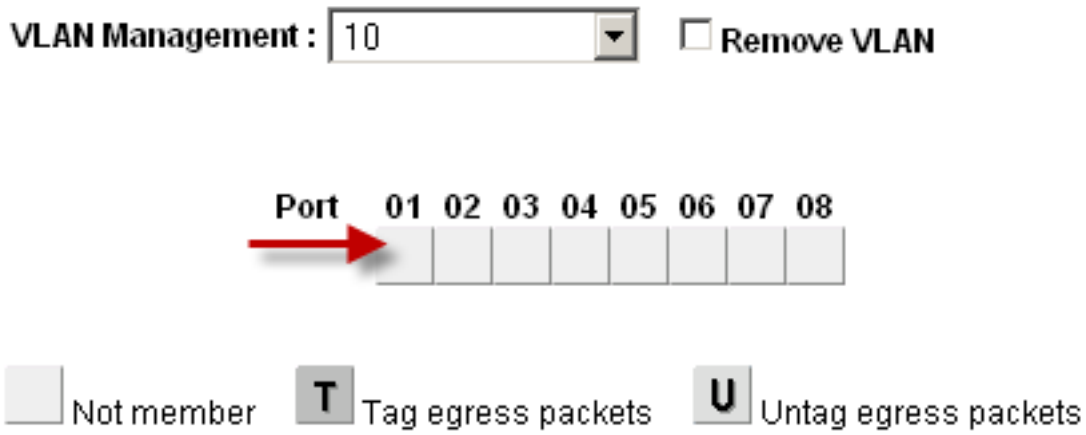


Fig. 100: Toggle VLAN Membership

Configure VLAN 10 membership

Figure *Configure VLAN 10 Membership* shows VLAN **10** configured as outlined in Table *table-netgear-gs108t-vlan-configuration*. The access ports on this VLAN are set to **untagged** while the trunk port is set to tagged.

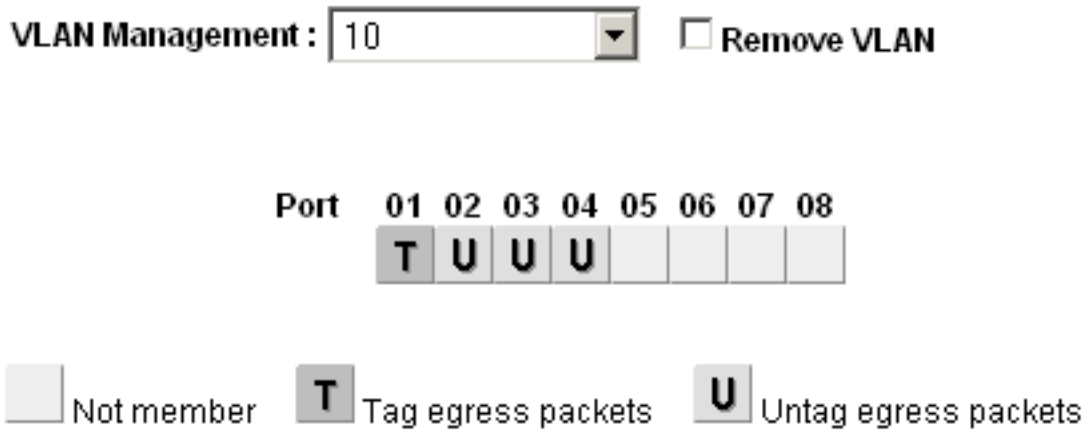


Fig. 101: Configure VLAN 10 Membership

Configure VLAN 20 membership

Select **20** from the VLAN Management drop down to configure the port memberships for VLAN **20**.

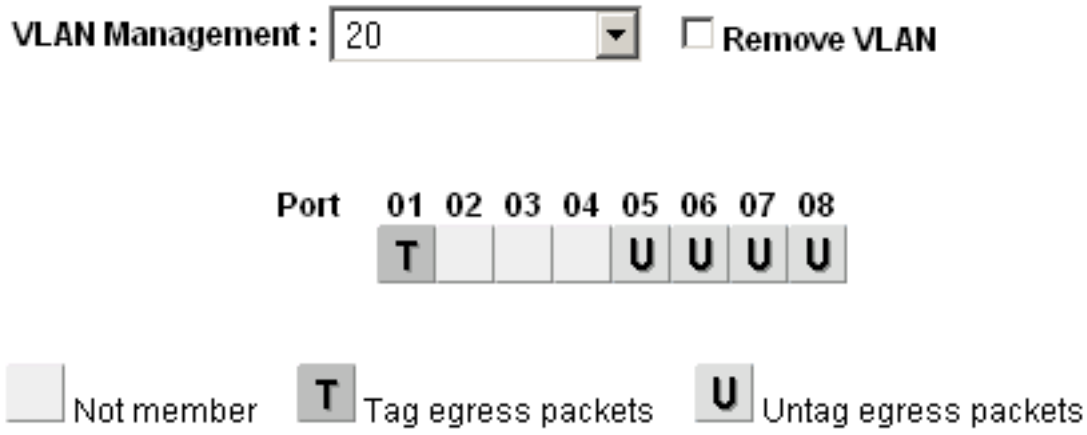


Fig. 102: Configure VLAN 20 Membership

Change PVID

On Netgear switches, in addition to the previously configured tagging settings, the PVID must also be configured to specify the VLAN used for frames entering a port:

- Select **PVID** from the VLAN Management drop down as shown in Figure *PVID Setting*.

  The default PVID setting is VLAN 1 for all ports as shown in Figure *Default PVID Configuration*.

- Change the PVID for each access port, but leave the trunk port and port used to access the switch management interface set to *1* .

  Figure *VLAN 10 and 20 PVID Configuration* shows the PVID configuration matching the port assignments shown in Table *Netgear GS108T VLAN Configuration*, with port 8 being used to access the switch management interface.

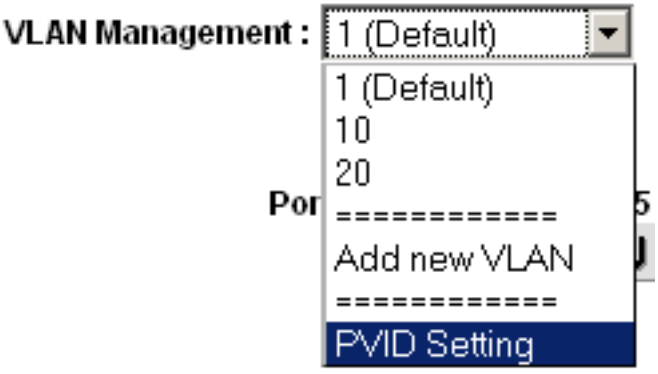


Fig. 103: PVID Setting

VLAN Management : PVID Setting

Port	PVID	Port	PVID	Port	PVID	Port	PVID
01	1	02	1	03	1	04	1
05	1	06	1	07	1	08	1

Fig. 104: Default PVID Configuration

VLAN Management : PVID Setting

Port	PVID	Port	PVID	Port	PVID	Port	PVID
01	1	02	10	03	10	04	10
05	20	06	20	07	20	08	1

Fig. 105: VLAN 10 and 20 PVID Configuration

- Apply changes when finished

Remove VLAN 1 configuration

By default, all ports are members of VLAN 1 with untagged egress frames. To remove VLAN 1 from the other ports:

- Select *1 (Default)* from the **VLAN Management** drop down
- Remove VLAN 1 from all ports except the one used to manage the switch and the trunk port, to avoid being disconnected.

  In this example, port 8 is used to manage the switch. When finished, the screen will look like Figure *Remove VLAN 1 Membership*.

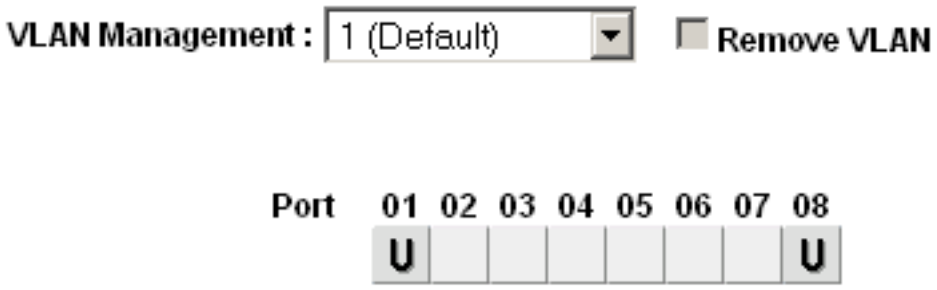


Fig. 106: Remove VLAN 1 Membership

- Apply changes when finished

Verify VLAN functionality

Configure VLANs on pfSense, including the DHCP server on the VLAN interfaces if needed. Plug systems into the configured access ports and test connectivity. If everything works as desired, continue to the next step. If things do not work as intended, review the tagging and PVID configuration on the switch, and the VLAN configuration and interface assignments on pfSense.

34.67.6 Dell PowerConnect managed switches

The management interface of Dell switches varies slightly between models, but the following procedure will accommodate most models. The configuration is quite similar in style to Cisco IOS.

First, create the VLANs:

```
console# config
console(config)# vlan database
console(config-vlan)# vlan 10 name dmz media ethernet
console(config-vlan)# vlan 20 name phones media ethernet
console(config-vlan)# exit
```

Next, setup a trunk port:

```
console(config)# interface ethernet 1/1
console(config-if)# switchport mode trunk
```

(continues on next page)

(continued from previous page)

```
console(config-if)# switchport allowed vlan add 1-4094 tagged
console(config-if)# exit
```

Finally, add ports to the VLANs:

```
console(config)# interface ethernet 1/15
console(config-if)# switchport allowed vlan add 10 untagged
console(config-if)# exit
```

34.68 Using the Shaper Wizard to Configure ALTQ Traffic Shaping

The easiest way to get started with traffic shaping is by using the wizard for the first time, which guides administrators through the shaper configuration process.

Tip: Due to the complexity of the shaper queues and rules, starting from scratch is quite complicated. If a firewall needs custom rules, step through the wizard and approximate the requirements, then make custom rules afterward.

Each step of the wizard sets up unique queues and rules that control what traffic is assigned into those queues. To configure everything manually, specify the WAN speed at the first screen, then click **Next** through all the remaining steps. The wizard requires options to be enabled on at least one step, but it does not matter which step.

Note: Completing the wizard and clicking **Finish** at the end will replace **all** existing shaper queues and floating rules created by the wizard, including those cloned from wizard rules, with the queues and rules from the new wizard configuration.

34.68.1 Choosing a Wizard

To get started with the Traffic Shaping Wizard, navigate to **Firewall > Traffic Shaper** and click the **Wizards** tab. This page displays a list of available traffic shaper wizards, including:

Multiple LAN/WAN Used when the firewall has one or more WANs and one or more LANs. This is the most common wizard and it covers most every scenario.

Dedicated Links Used when specific LAN+WAN pairings should be accounted for in the shaper configuration.

34.68.2 Starting the Wizard

Each wizard name is followed by the filename of the wizard, which is a link. Click the link to start the wizard. This example uses the **Multiple LAN/WAN** wizard, so click `traffic_shaper_wizard_multi_all.xml`.

Next, the wizard starts and the first step prompts for the number of WAN and LAN type connections on the firewall, as in Figure *Entering the Interface Count*.

- Enter the number of WAN-type connections on the firewall. These are connections with a gateway configured on the interface, or dynamic WAN type interfaces such as DHCP or PPPoE
- Enter the number of LAN type connections. These are local network interfaces without a gateway on the interface

- Click **Next** to proceed with the next step

In this example the firewall only has one WAN and one LAN interface.

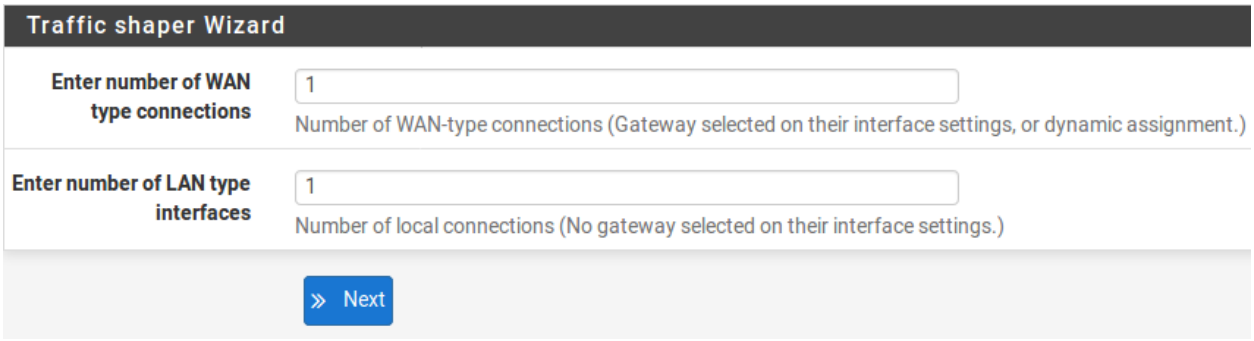


Fig. 107: Entering the Interface Count

34.68.3 Networks and Speeds

This step, shown in Figure *Shaper Configuration*, defines the network interfaces that will be the inside and outside from the point of view of the shaper, along with the **Download** and **Upload** speeds for a given WAN. When the firewall has more than one interface of a given type, the wizard displays multiple sections on the page to handle each one individually.

In addition to the interfaces and their speeds, select an ALTQ **Scheduler** (*ALTQ Scheduler Types*) for the WAN(s) and LAN(s). Use the same scheduler on every interface.

Depending on the connection type, the true link speed may not be the actual usable speed. In the case of PPPoE, the circuit has not only PPPoE overhead, but also overhead from the underlying ATM network link being used in most PPPoE deployments. By some calculations, between the overhead from ATM, PPPoE, IP, and TCP, the circuit may lose as much as 13% of the advertised link speed. When in doubt of what to set the speed to, be conservative. Reduce by 10-13% and work it back up to larger values. If the firewall has a 3Mbit/s line, set it for about `2.7` Mbit/s and then test. The speed on the resulting parent queue can be edited later to adjust the bandwidth. If it has a low value, the connection will be maxed out at exactly the defined speed. Nudge it up higher until the firewall no longer sees any performance gains.

Interface speeds can be specified in *Kbit/s* , *Mbit/s* , or *Gbit/s* but use the same units on every page.

- Choose an **Interface** and **Scheduler** for each LAN-type interface (e.g. *LAN*, *PRIQ*)
- Choose an **Interface** and **Scheduler** for each WAN-type interface (e.g. *WAN*, *PRIQ*)
- Define the **Upload** speed and units for each WAN-type interface (e.g. `1`, *Mbit/s*)
- Define the **Download** speed and units for each WAN-type interface (e.g. `10`, *Mbit/s*)
- Click **Next** to proceed with the next step

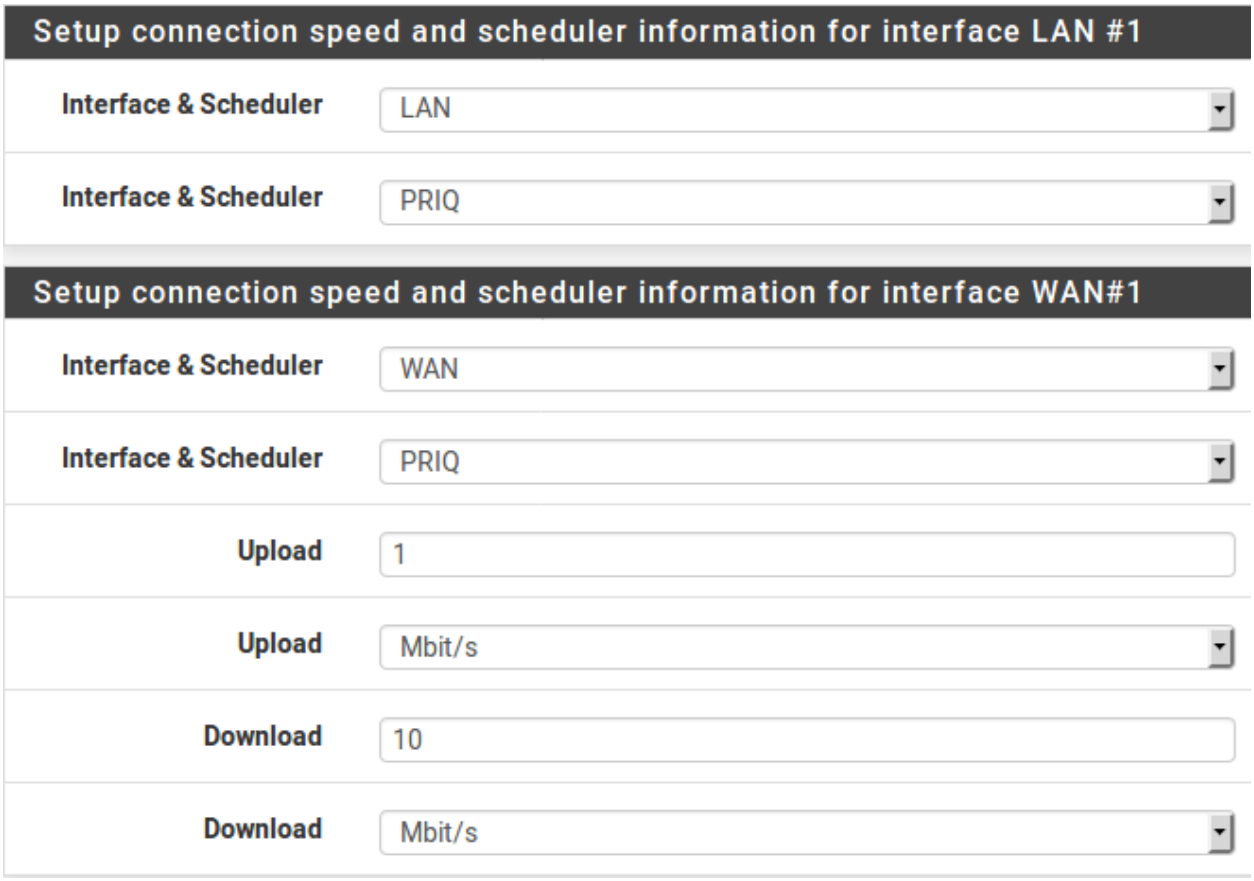


Fig. 108: Shaper Configuration

34.68.4 Voice over IP

The wizard contains several options for handling VoIP call traffic, shown in Figure *Voice over IP*. Prioritizing Voice over IP traffic sets up queues and rules to give priority to VoIP calls and related traffic. This behavior can be fine-tuned by the other settings on this step of the wizard.

Enable A checkbox to enable the VoIP settings on this step. When unchecked, the options are disabled and these queues and rules will not be added by the wizard.

Provider There are a few well-known providers including *Vonage*, *Voicepulse*, *PanasonicTDA*, and *Asterisk* servers. If the VoIP provider for this site is not in the list, choose *Generic*. This choice sets up rules based on the ports and protocols known to be used by these providers, rather than matching by address.

Note: This choice matches based on SIP and RTP ports, among others, therefore it can match traffic from other sources as well if they use the same ports as the selected service.

Upstream SIP Server The IP of the upstream PBX or SIP trunk, or an alias containing the IP addresses or networks for the SIP trunk(s). When set, this overrides the **Provider** field and will instead match traffic based on these addresses.

Note: This choice matches all UDP traffic to and from the specified address(es). In most cases this is OK, but if there are other Non-VoIP UDP-based services on the same remote address, it could match that traffic as well. Such cases are rare, however, so this option tends to be more reliable than matching by port.

WAN Connection Upload The amount of upload bandwidth to guarantee for VoIP devices. This will vary based on how many VoIP devices are on the network and how much bandwidth each session requires. This setting is used by HFSC and CBQ, and should be left blank for PRIQ.

Note: The bandwidth reservation for a service such as VoIP cannot exceed 30% of the available bandwidth on the link. For example, on a 10Mbit/s link, the shaper cannot reserve more than 3Mbit/s.

LAN Connection Download The amount of download bandwidth to guarantee for VoIP devices. This setting is used by HFSC and CBQ, and should be left blank for PRIQ.

Note: The best practice is to use the **remote** SIP trunk or PBX address because otherwise the shaper may not be able to match traffic properly. For example, using the IP addresses of phones the shaper may only match traffic in one direction, or not at all. This is due to the way the shaper matches traffic with floating rules in an outbound direction. NAT applies before traffic is matched when exiting a WAN, so the shaper rules cannot match outbound connections based on local private IP addresses.

To use these options:

- Check **Prioritize Voice over IP traffic**
- Pick **ONE** of the following:
  - Choose a **Provider** from the list **OR**
  - Enter an **Upstream SIP Server** address or alias containing a **remote** SIP trunk or PBX

- Leave **Upload** and **Download** blank if using PRIQ, otherwise enter an appropriate **Upload** or **Download** value for each connection
- Click **Next** to proceed with the next step

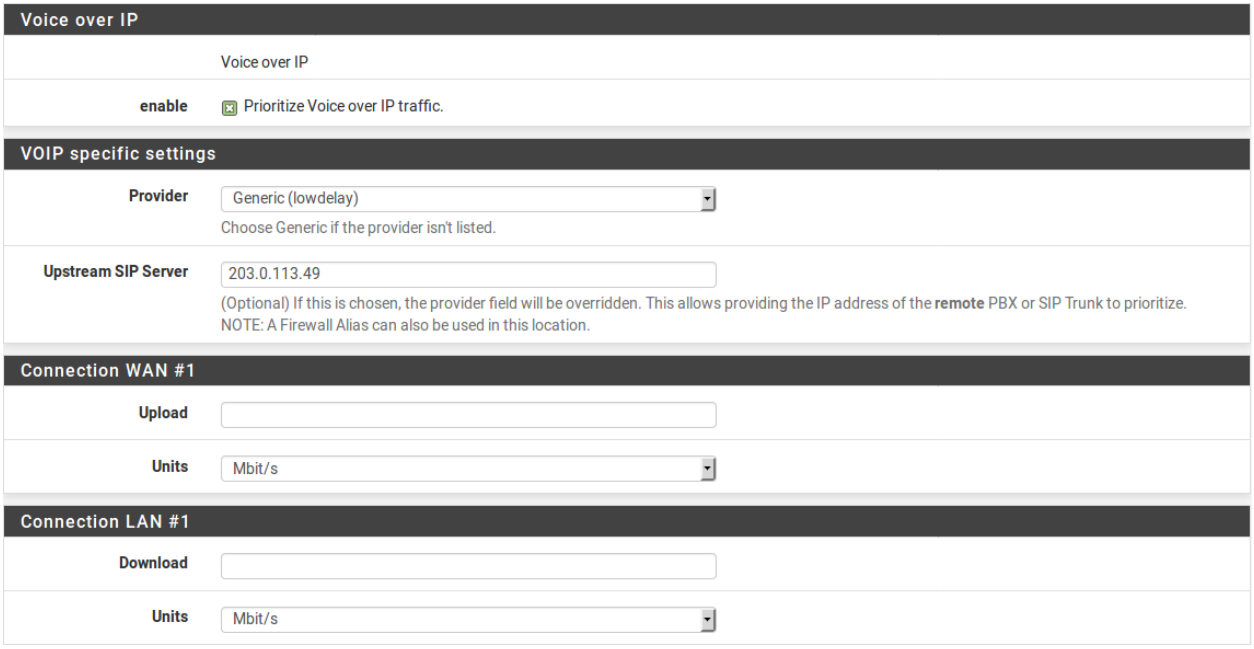


Fig. 109: Voice over IP

34.68.5 Penalty Box

The penalty box, depicted in Figure *Penalty Box*, is a place to relegate misbehaving users or devices that would otherwise consume undesirable amounts of bandwidth. These devices are assigned a hard bandwidth cap which they cannot exceed.

Enable A checkbox to enable the Penalty Box settings on this step. When unchecked, the options are disabled and these queues and rules will not be added by the wizard.

Address The IP address to penalize, or an alias containing multiple addresses to penalize.

Bandwidth The amount of bandwidth that **Address** can consume, at most.

To use these options:

- Check **Penalize IP or Alias**
- Enter an IP address or Alias in the **Address** box
- Enter the **Bandwidth** limit
- Choose the correct units for the **Bandwidth** limit
- Click **Next** to proceed with the next step

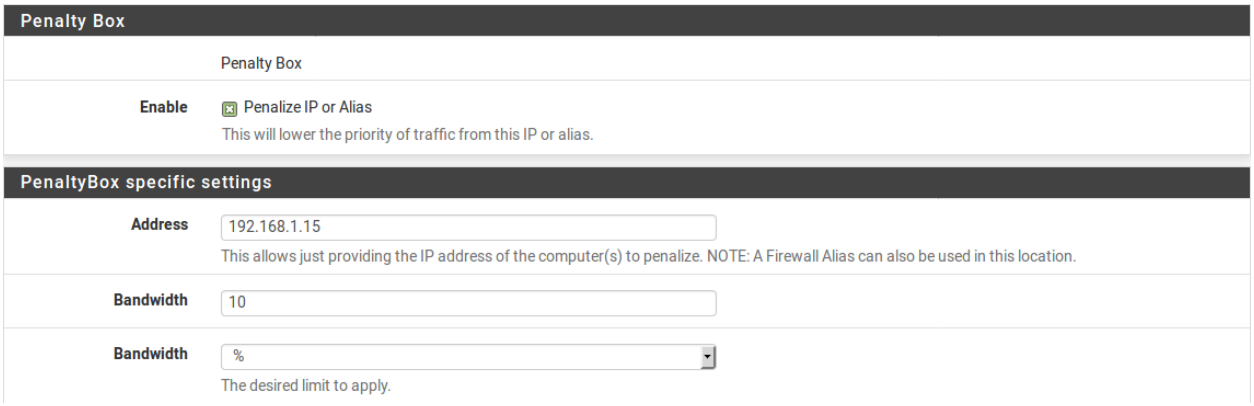


Fig. 110: Penalty Box

34.68.6 Peer-to-Peer Networking

The next step, shown in Figure *Peer-to-Peer Networking*, sets controls for many Peer-to-Peer (P2P) networking protocols. By design, P2P protocols will utilize all available bandwidth unless limits are put in place. If P2P traffic will be present on a network, the best practice is to ensure it will not degrade other traffic.

Note: P2P protocols deliberately attempt to avoid detection. Bittorrent is especially guilty of this behavior. It often utilizes non-standard or random ports, or ports associated with other protocols. Identifying all P2P traffic can be difficult or impossible.

Enable A checkbox to enable the P2P traffic settings on this step. When unchecked, the options are disabled and these queues and rules will not be added by the wizard.

Peer-to-Peer Catch All Causes any unrecognized traffic to be assumed as P2P traffic, and such traffic will have its priority lowered accordingly.

Bandwidth The amount of bandwidth that unclassified traffic can consume, at most, when P2P Catch All is active.

> **Warning:** This option effectively takes over the **Default** traffic shaping queue and lowers its priority. When this option is active, it is critical for all legitimate traffic to be matched by rules that set a priority higher than the priority of the P2P catch all queue.
>
> The **Raise / Lower Other Applications** step of the wizard can help here, but ultimately accomplishing this task frequently requires additional manual rules.

Enable/Disable specific P2P protocols These options identify various known P2P protocols. The firewall will assign ports and protocols associated with each enabled option as P2P traffic.

To use the options in this step:

- Check **Lower priority of Peer-to-Peer traffic**
- Optionally enable the **p2p Catch All** feature
  - Enter the **Bandwidth** limit for **p2p Catch all**, if enabled
  - Choose the correct units for the **Bandwidth** limit
- Select protocols for the firewall to classify as P2P traffic

- Click **Next** to proceed with the next step

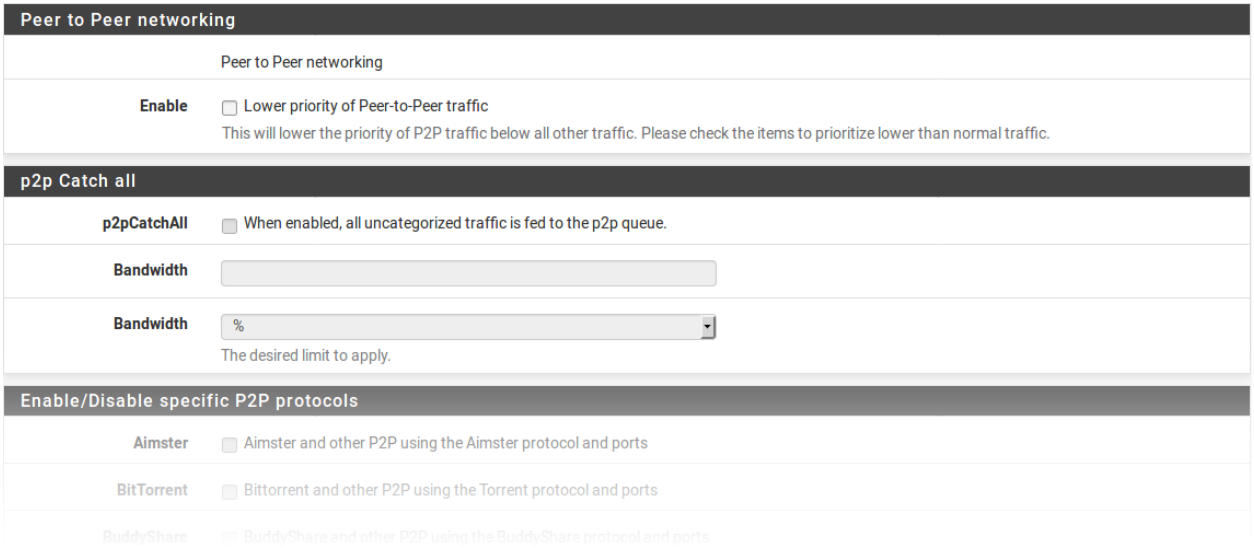


Fig. 111: Peer-to-Peer Networking

34.68.7 Network Games

Online games typically rely on low latency for acceptable player experiences. If a user on the network attempts to download large files or game patches while playing, that traffic can easily drown out the packets associated with the game itself and cause lag or disconnections. If the firewall gives gaming traffic priority, it can ensure that traffic will be delivered first and fastest.

Enable A checkbox to enable the gaming traffic settings on this step. When unchecked, the options are disabled and these queues and rules will not be added by the wizard.

Enable/Disable specific game consoles and services These options match traffic for entire game consoles or online services which use common ports and protocols across all, or at least a majority, of their games.

Enable/Disable specific games These options match traffic for specific games which deviate from the general categories in the previous section.

Tip: To prioritize a game that is not listed, check any other game from the list so that the wizard will create the queues and rules to use as a reference base. After completing the wizard, edit the resulting rules to match the unlisted game.

To use the options in this step:

- Check **Prioritize network gaming traffic**
- Select any games consoles on the network from the list in **Enable/Disable specific game consoles and services**
- Select any games on the network from the list in **Enable/Disable specific games**
- Click **Next** to proceed with the next step

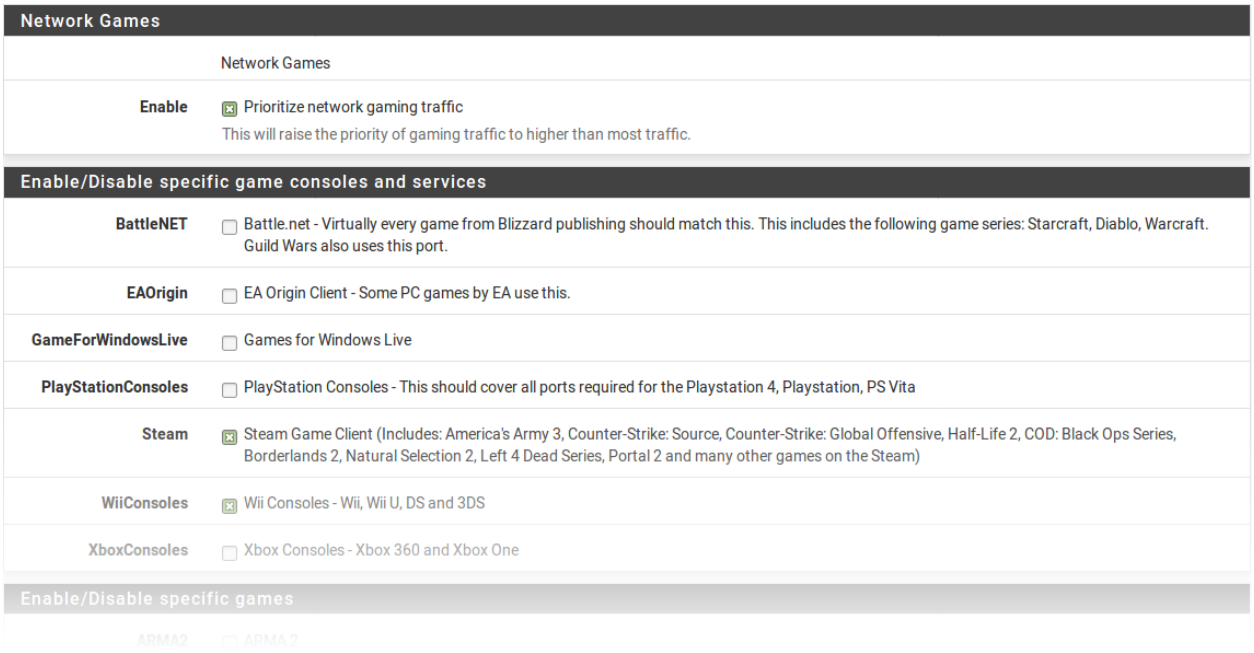


Fig. 112: Network Games

34.68.8 Raising or Lowering Other Applications

The last configuration screen of the shaper wizard, seen in Figure *Raise or Lower Other Applications*, lists a number of other commonly available applications and protocols.

The needs of a particular network dictate how the firewall should handle each protocol. For example, in a corporate environment management may want to lower the priority of non-interactive traffic such as e-mail where a reduction in speed is not usually noticed by users, and they may also want to raise the priority of interactive services like RDP where poor performance is an impediment for employees. In a home, multimedia streaming may be more important, and other services can have their priority lowered by the shaper.

Tip: As with other steps of this shaper wizard, if a protocol is not listed, select a similar protocol and then adjust the rules after completing the wizard.

Enable A checkbox to enable the settings on this step. When unchecked, the options are disabled and these queues and rules will not be added by the wizard.

Protocol Categories Each section contains well-known protocols, grouped by their general function.

There are more than 40 protocols to choose from, and each can be given a *Higher priority*, *Lower priority*, or left at the *Default priority*.

Tip: If **p2pCatchAll** is active, we strongly recommend using this step to ensure that these other protocols are recognized and treated normally, rather than penalized by the default p2pCatchAll rule.

To use the options in this step:

- Check **Other networking protocols**
- Locate specific protocols in the list to alter priority.

- For each protocol, choose one of *Higher priority*, *Lower priority*, or leave it at the *Default priority*.
- Click **Next** to proceed with the next step

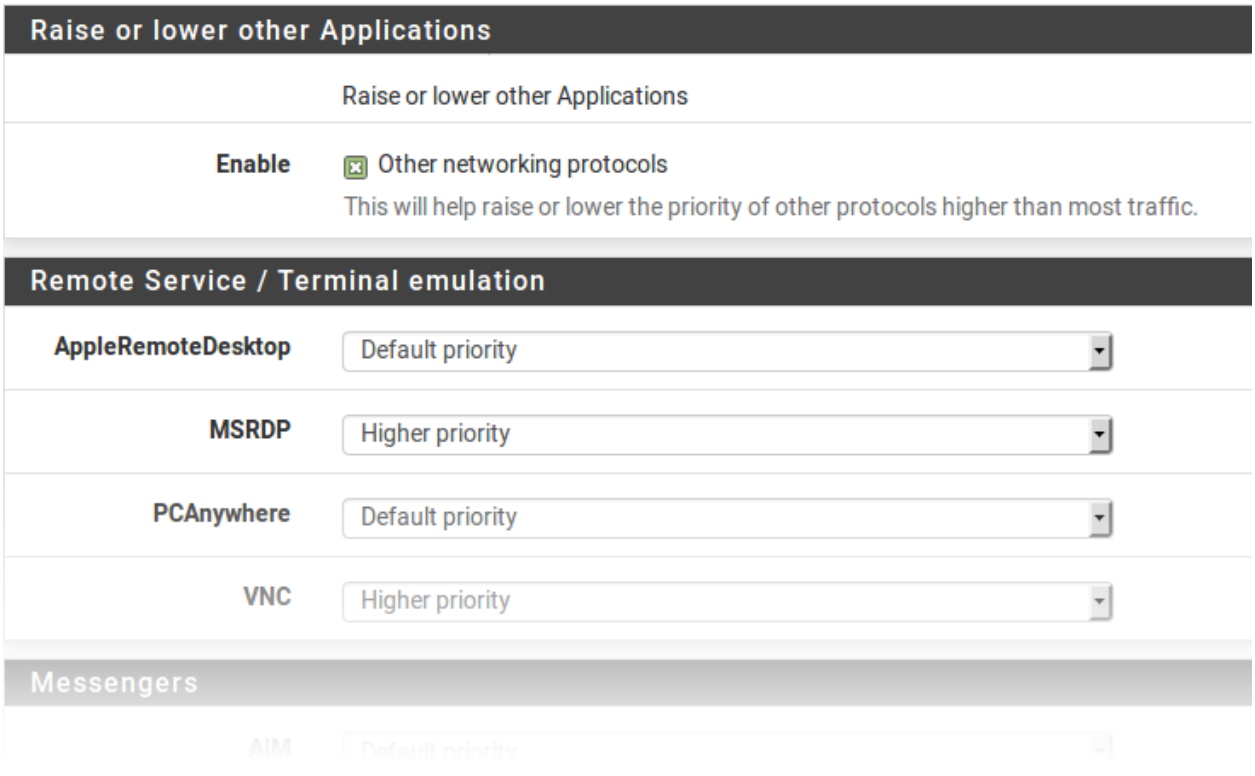


Fig. 113: Raise or Lower Other Applications

34.68.9 Finishing the Wizard

Click **Finish** to complete the wizard. The firewall will then create all of the rules and queues for enabled options, and then it will reload the ruleset to activate the new traffic shaper settings.

Due to the firewall operating in a stateful manner, the firewall can only apply changes in traffic shaping to new connections. In order for the new traffic shaping settings to be fully active on all connections, clear the states.

To reset the state table contents:

- Navigate to **Diagnostics > States**
- Click the **Reset States** tab
- Check **Reset the firewall state table**
- Click **Reset**

34.68.10 Shaper Wizard and IPv6

The shaper wizard creates rules for IPv4 traffic only. Rules can be manually adjusted or cloned and set for IPv6.

34.69 Virtualizing pfSense with VMware vSphere / ESXi

This article is about building a pfSense® virtual machine on vSphere / ESXi. Article explains how to install any major pfSense software version on VMware vSphere versions 5.x and 6.x. The article does not cover how to install vSphere or how to configure pfSense software to do any of the many amazing things it can. A basic, working, pfSense virtual machine will exist by the end of this document.

Note: If the pfSense firewall will be running as a perimeter firewall for an organization and the "attack surface" should be minimized, many will say it is preferable to run it unvirtualized on stand-alone hardware. That is a decision for the user and/or organization to make, however. Now back to the topic.*

We're going to start at the point where we have a vanilla ESXi install and have connected to it using the vSphere client. If other VMs are already running on ESXi, then it is not likely necessary to follow the networking steps too closely. However, we recommend skimming through it to see what is suggested before building the pfSense virtual machine part.

34.69.1 Assumptions

- vSphere host is up and running
- The reader has an understanding of network addressing
- The pfSense software installation `.iso` image is in a datastore

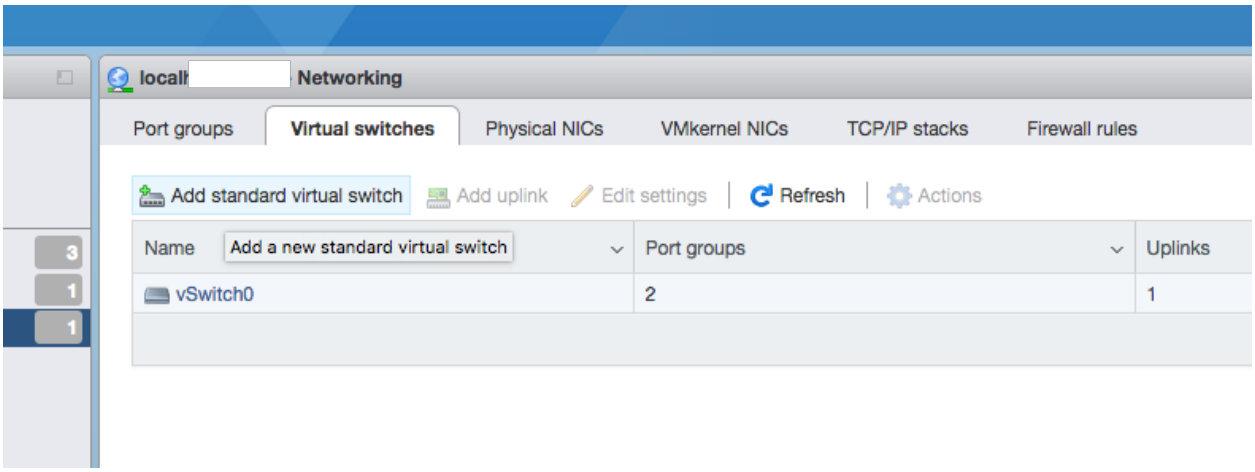
34.70 Installing pfSense Software on vSphere 6.x using vSphere web client

The following steps include the necessary vSphere web client configuration required to get pfSense VM running. After getting to the pfSense setup step, switch to the guide for vSphere client below.

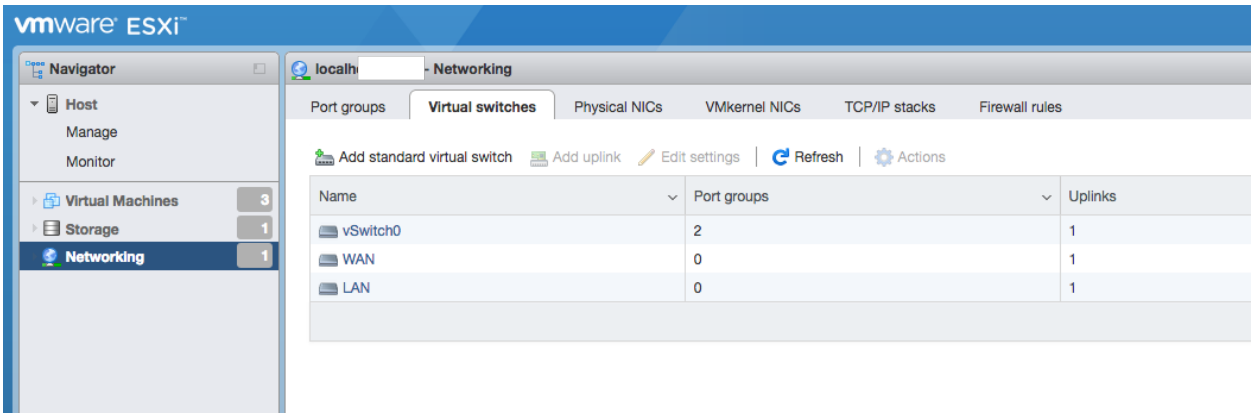
34.70.1 Basic vSphere web client networking setup

Before creating a new VM in vSphere web client, create two virtual switches and two port groups. First, create Virtual switches for WAN and LAN and after that two port groups for the WAN and LAN.

From the vSphere web client navigator, click on Networking and then click on Virtual switches tab. From there, click on "Add a new standard virtual switch".

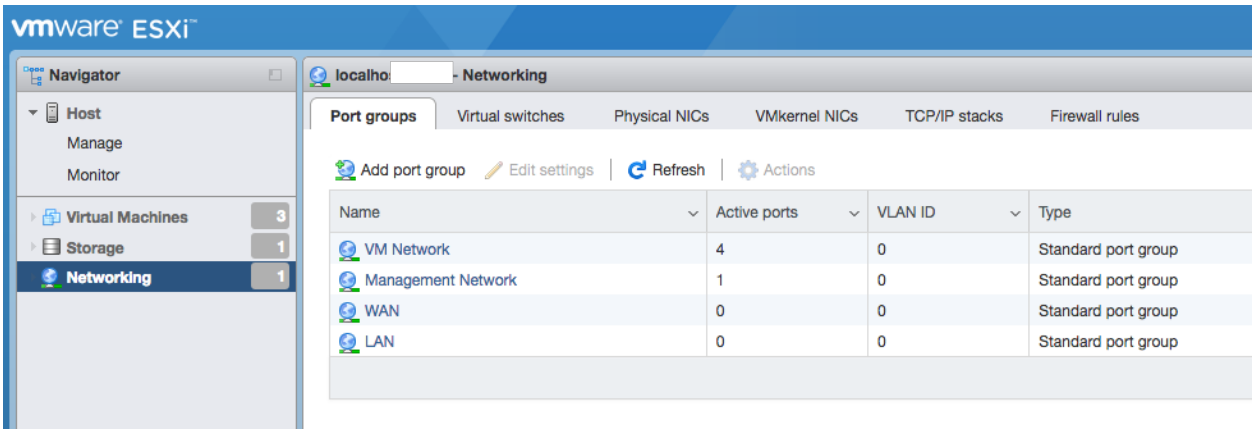


Add two Virtual switches, one for WAN and another for LAN. For uplink select two separate available ports.



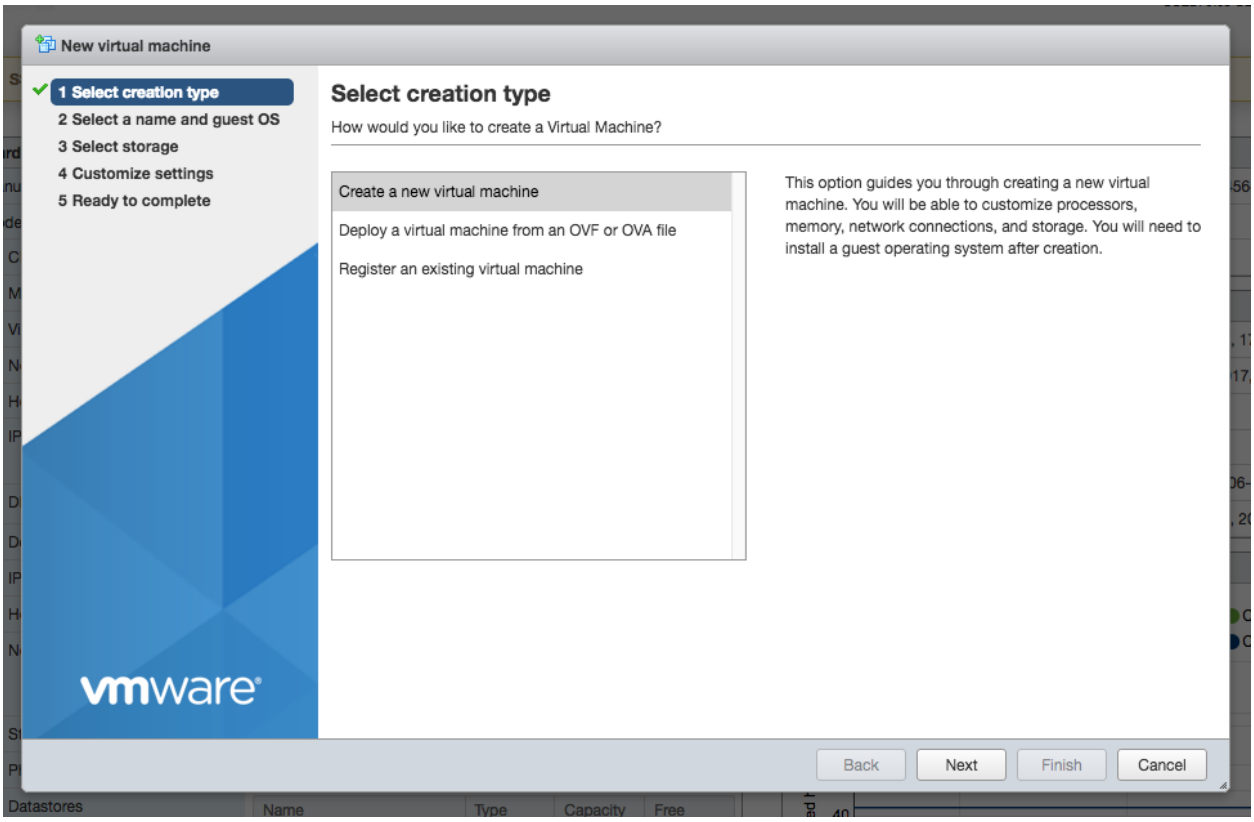
34.70.2 Creating port groups

After creating Virtual switches, click on Port groups tab. On the Port groups tab click on “Add port group”. Add WAN and LAN port groups, each using WAN and LAN switches respectively.

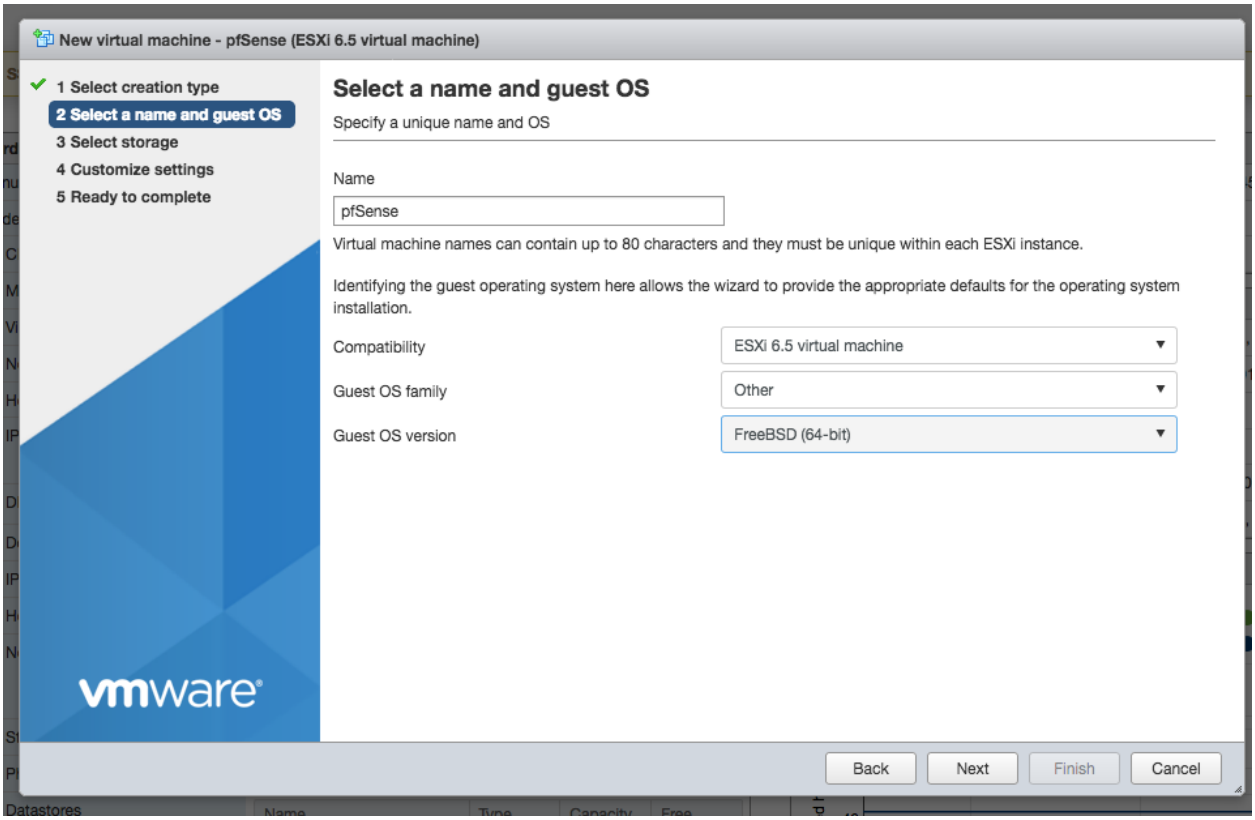


34.70.3 Creating a pfSense VM

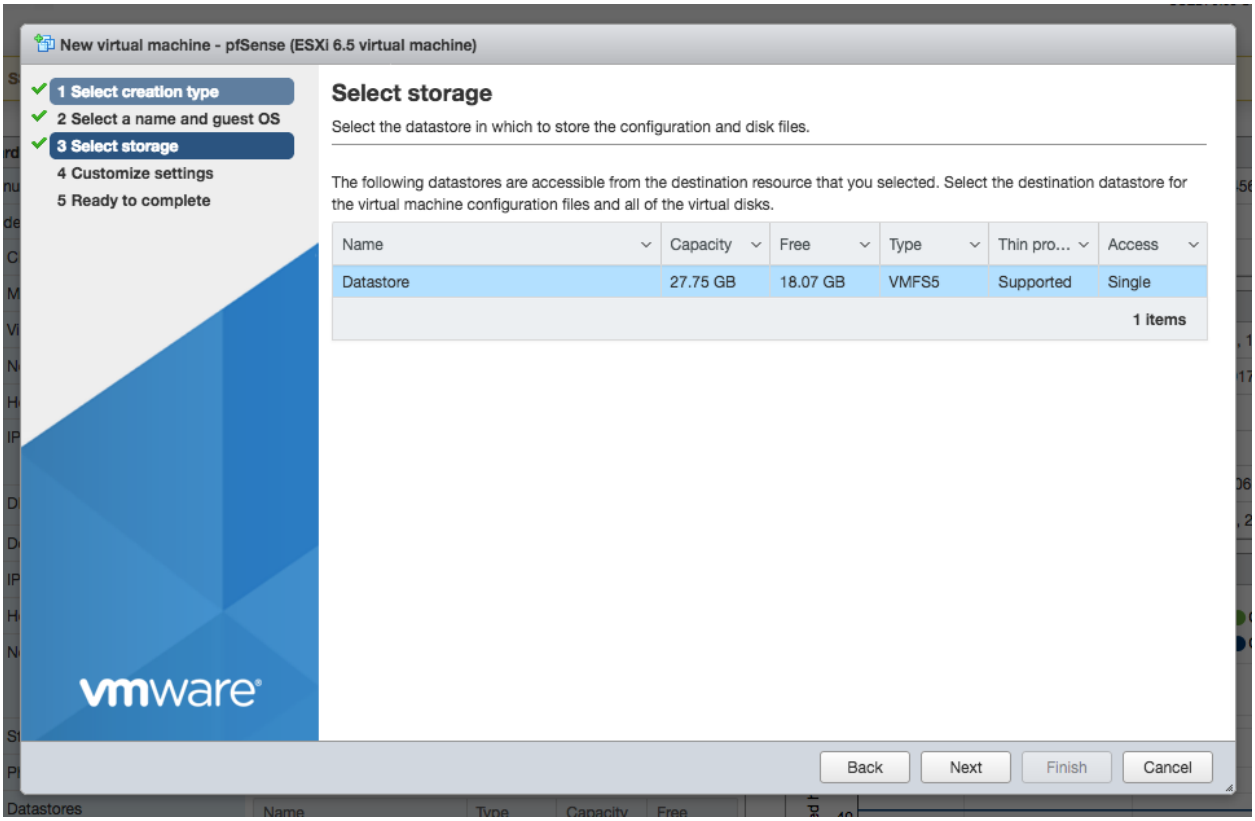
Now that the networking part is done, we continue to create a virtual machine. From the dashboard click on "Create/Register VM". On the first wizard screen select "Create a new virtual machine".



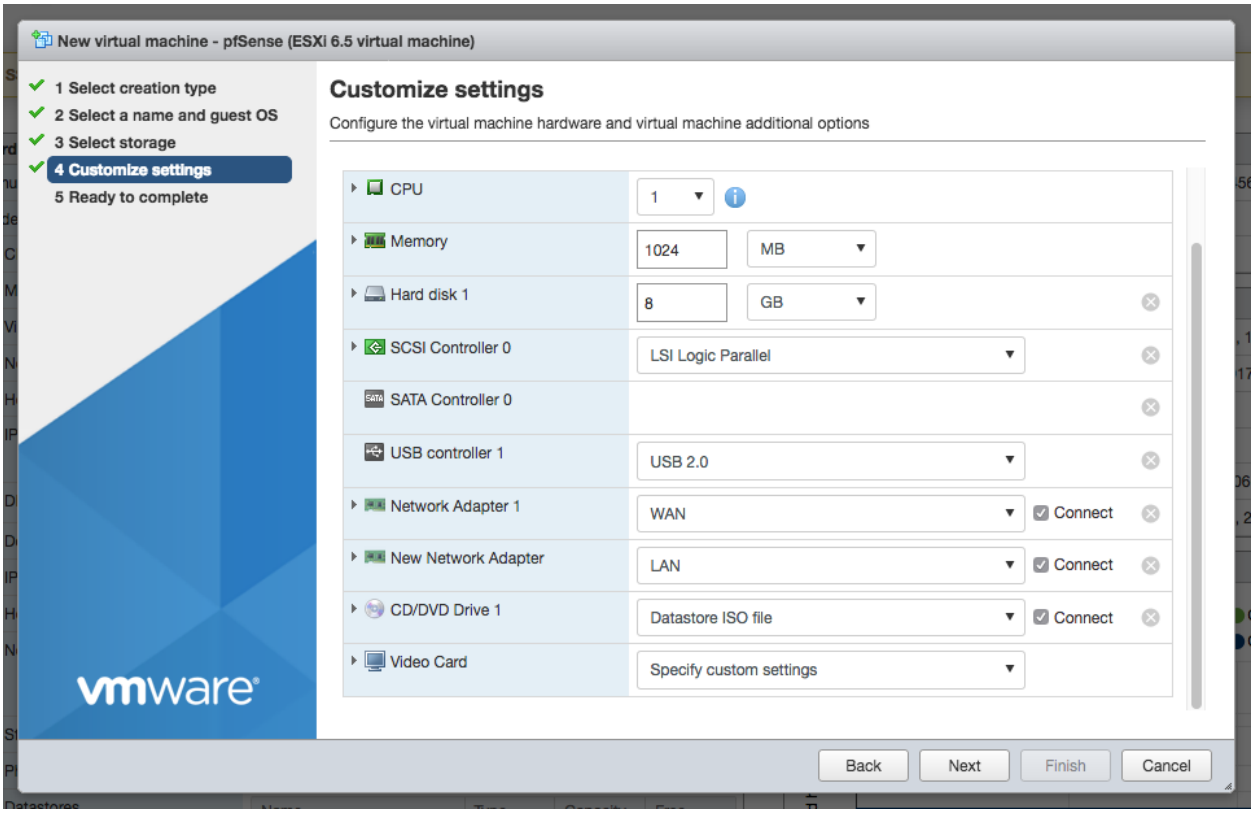
On the second page of the wizard, enter a name for the VM and select correct Guest OS version.



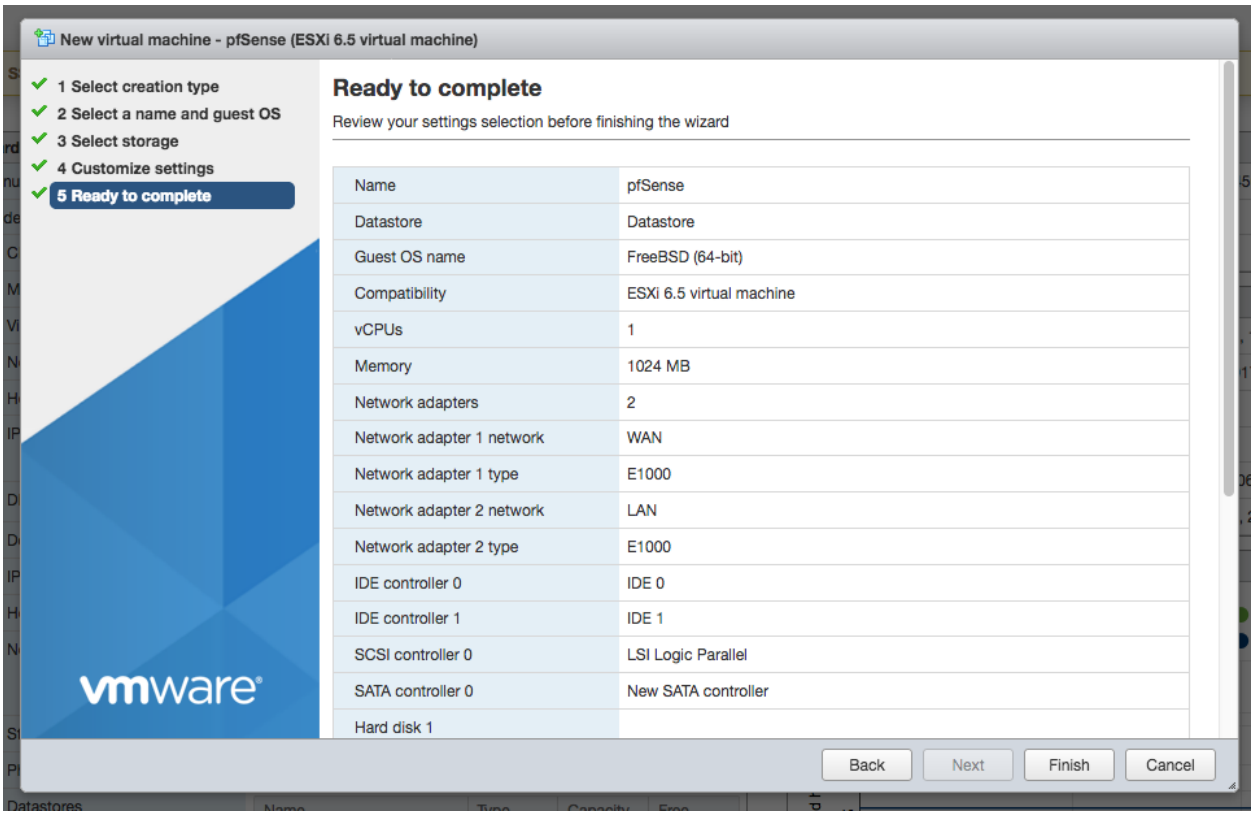
On the third page of the wizard, select the datastore where the hypervisor will keep the VM.



On wizard page four, add another Network Adapter and select the WAN and LAN port groups for each of the network adapters. Modify other virtual machine settings as needed. For best performance, use VMXNET 3 type of adapters instead of E1000. However, VMXNET 3 interfaces require manual interface assignment with the first boot. This guide uses the E1000 adapter type.

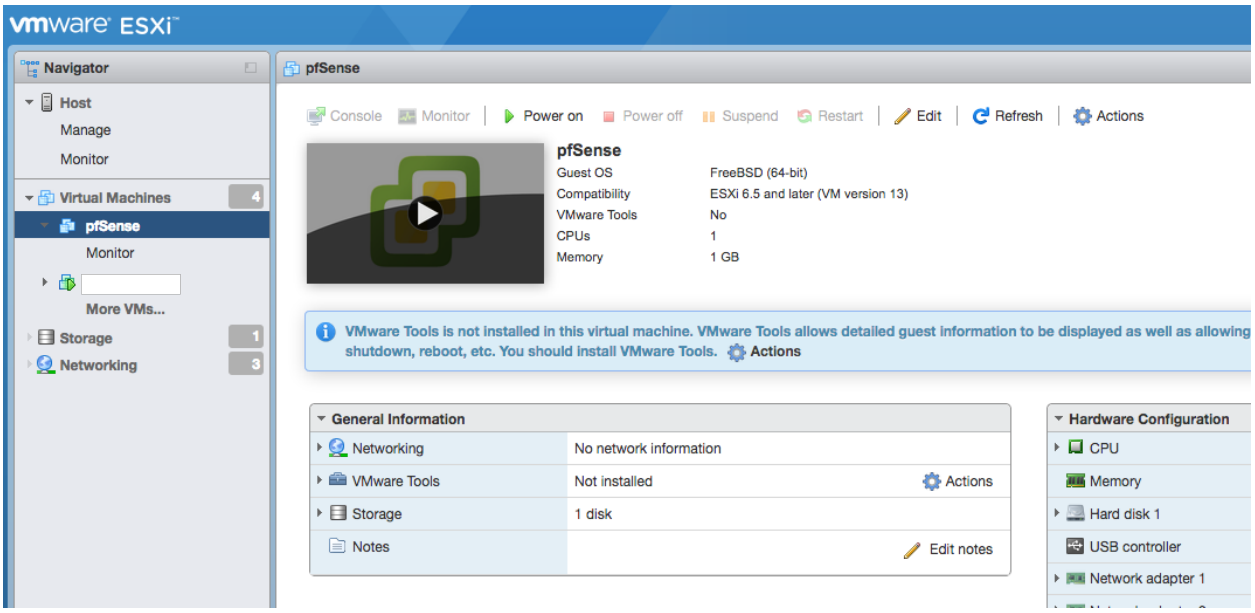


On the final wizard screen confirm the settings and click finish.



34.70.4 pfSense software installation

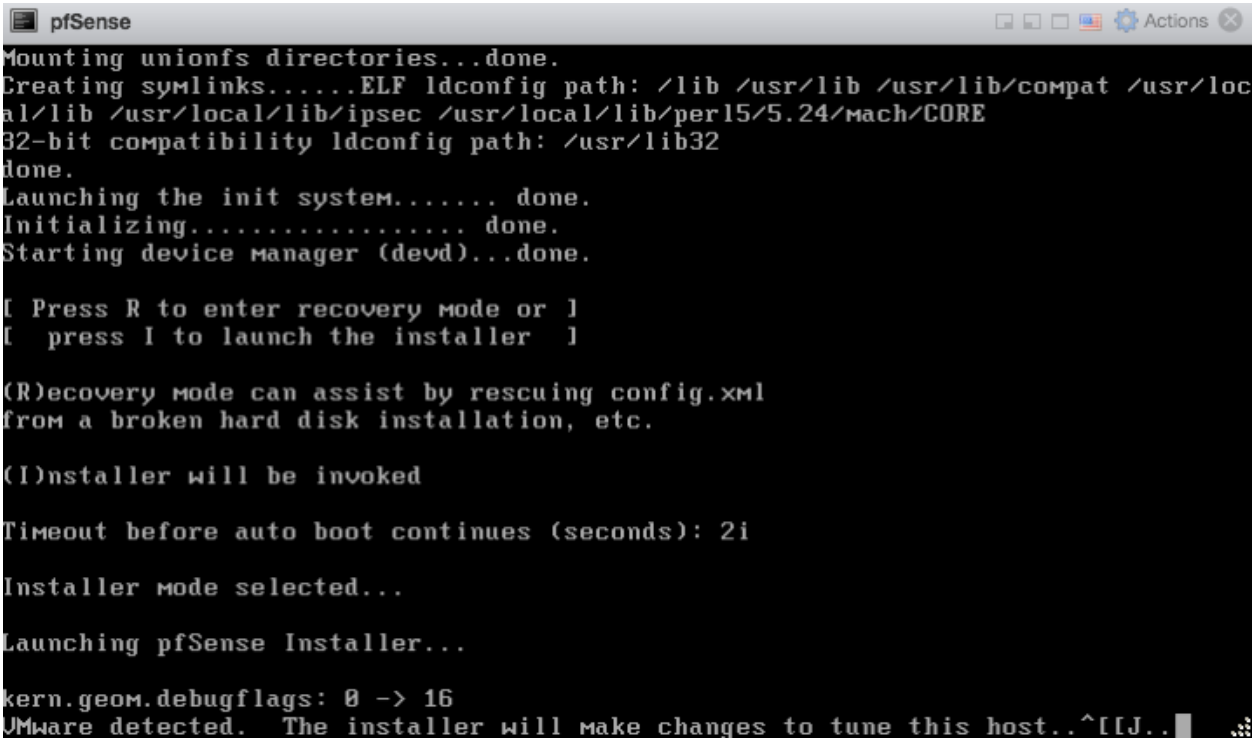
Once the pfSense virtual machine is created, under vSphere web client navigator click on "Virtual Machines" and select the newly created VM.



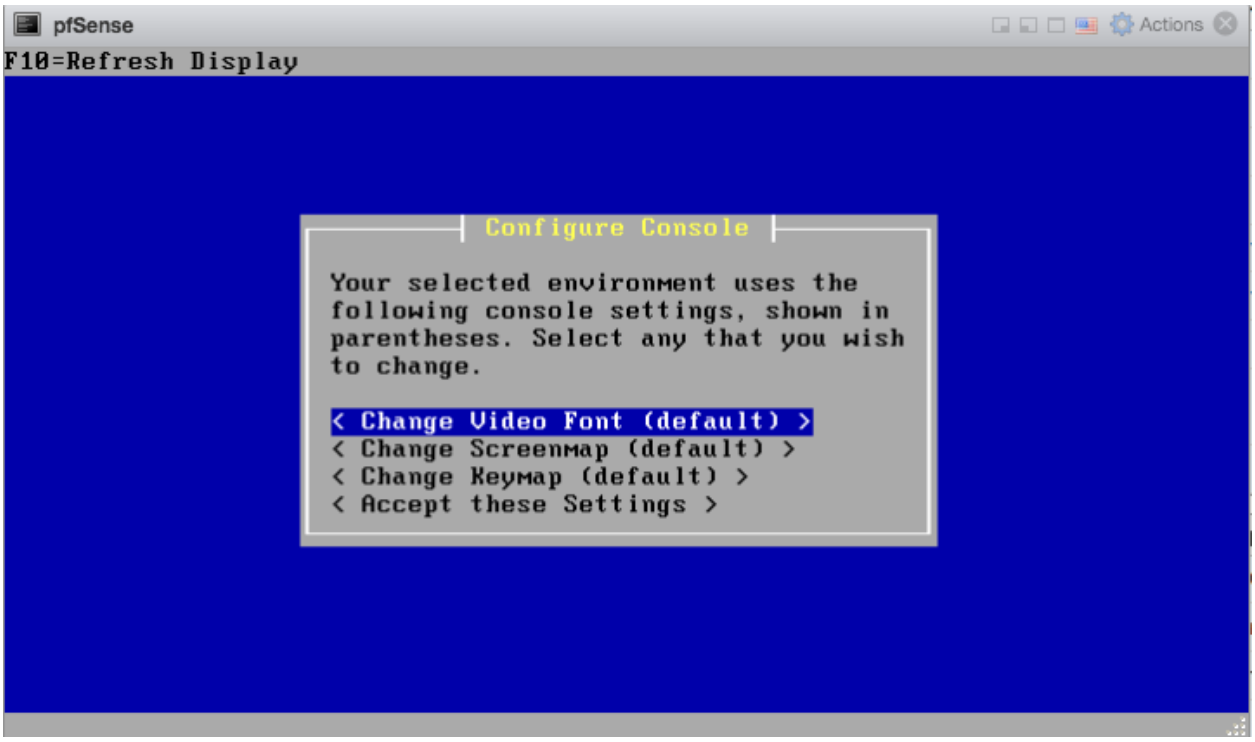
Power on the virtual machine.



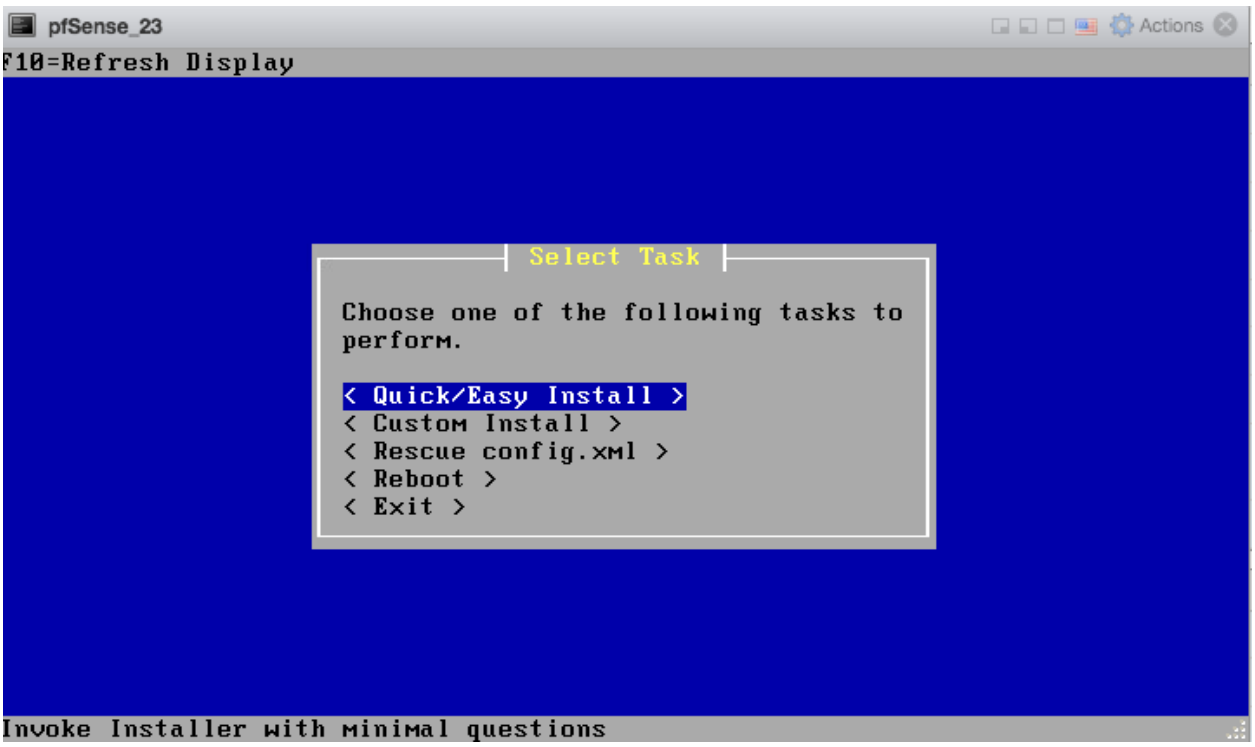
On the next screen, press "I" to invoke installer mode.



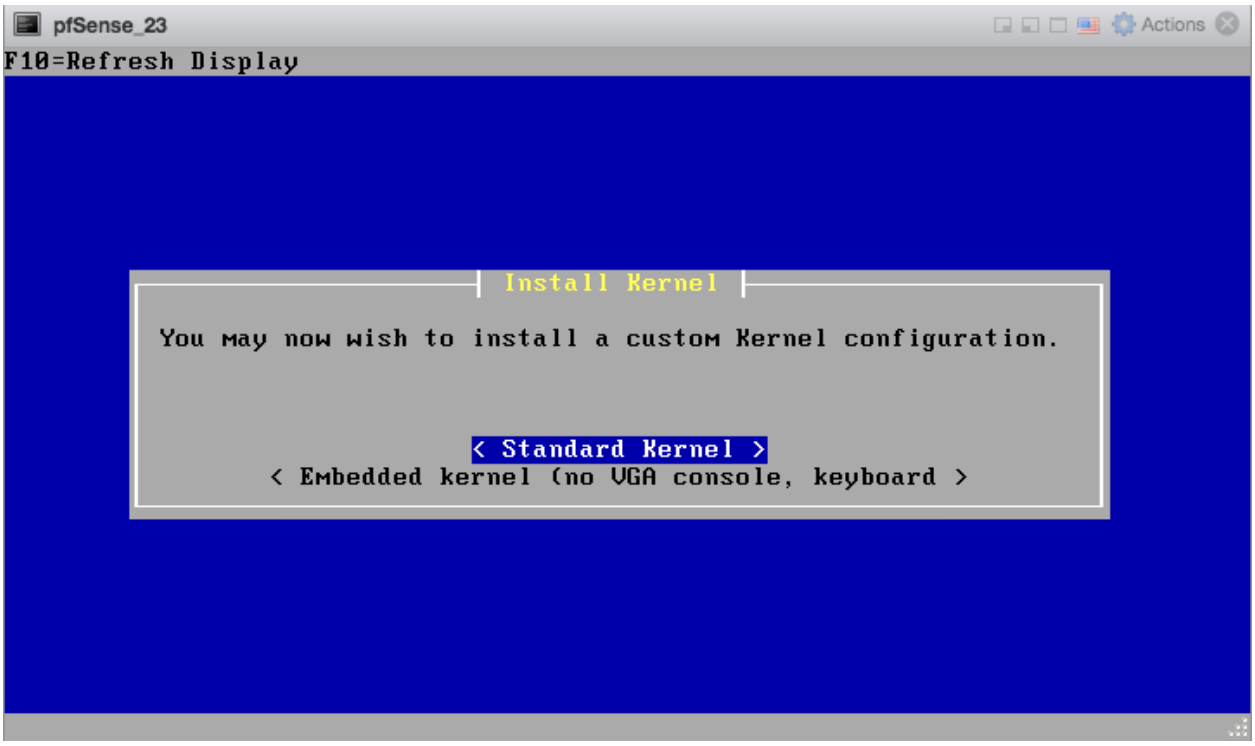
The pfSense software installer starts automatically. Select "Accept these settings"



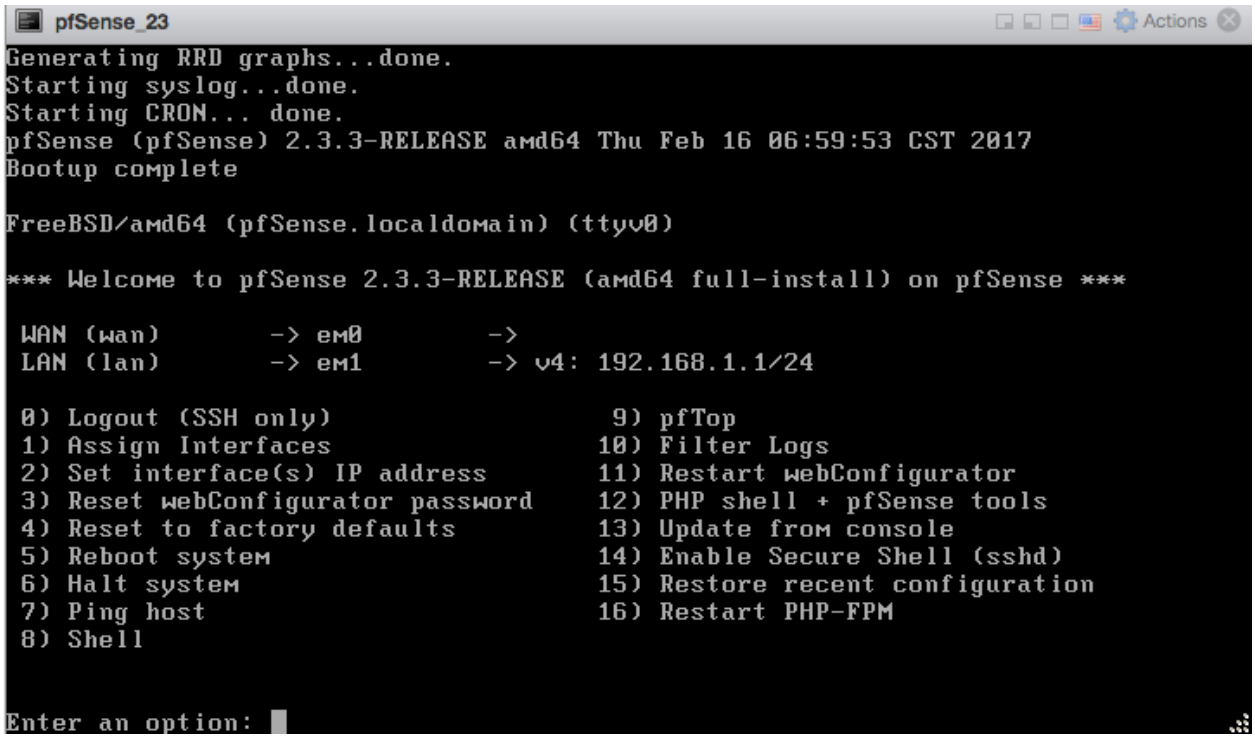
On the following screen choose “Quick/Easy Install” after which installation starts.



When prompted, select “Standard Kernel”.

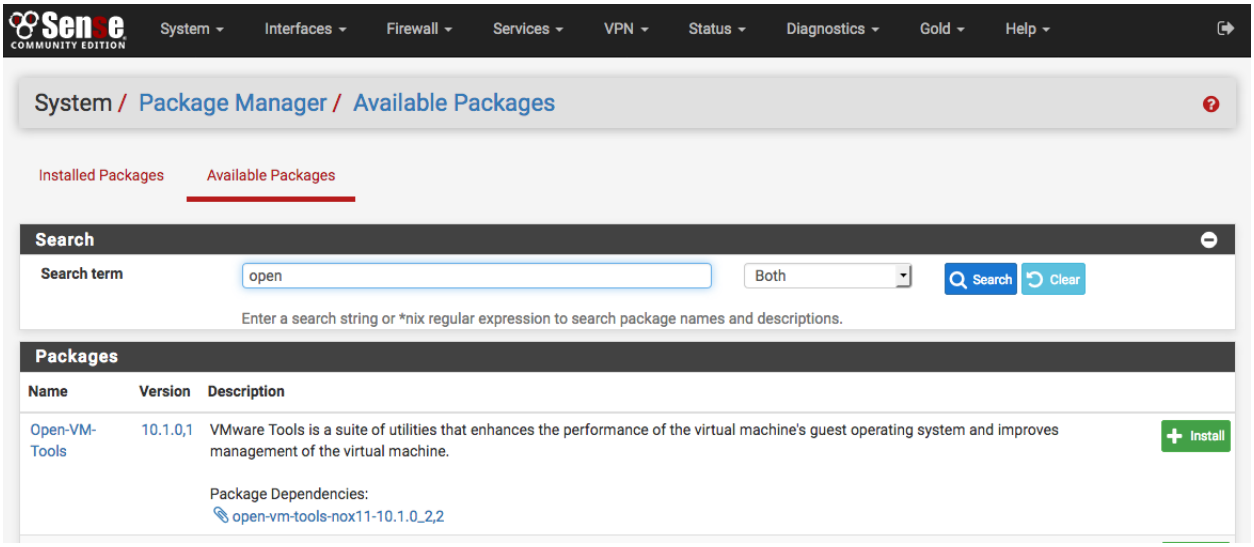


After that installation completes and pfSense software boots up for the first time.



34.70.5 Installing Open-VM-Tools

Once the pfSense installation is complete, upon first boot install the Open-VM-Tools. Reboot is not necessary afterwards, however make sure the Open-VM-Tools service is running under Status > Services.



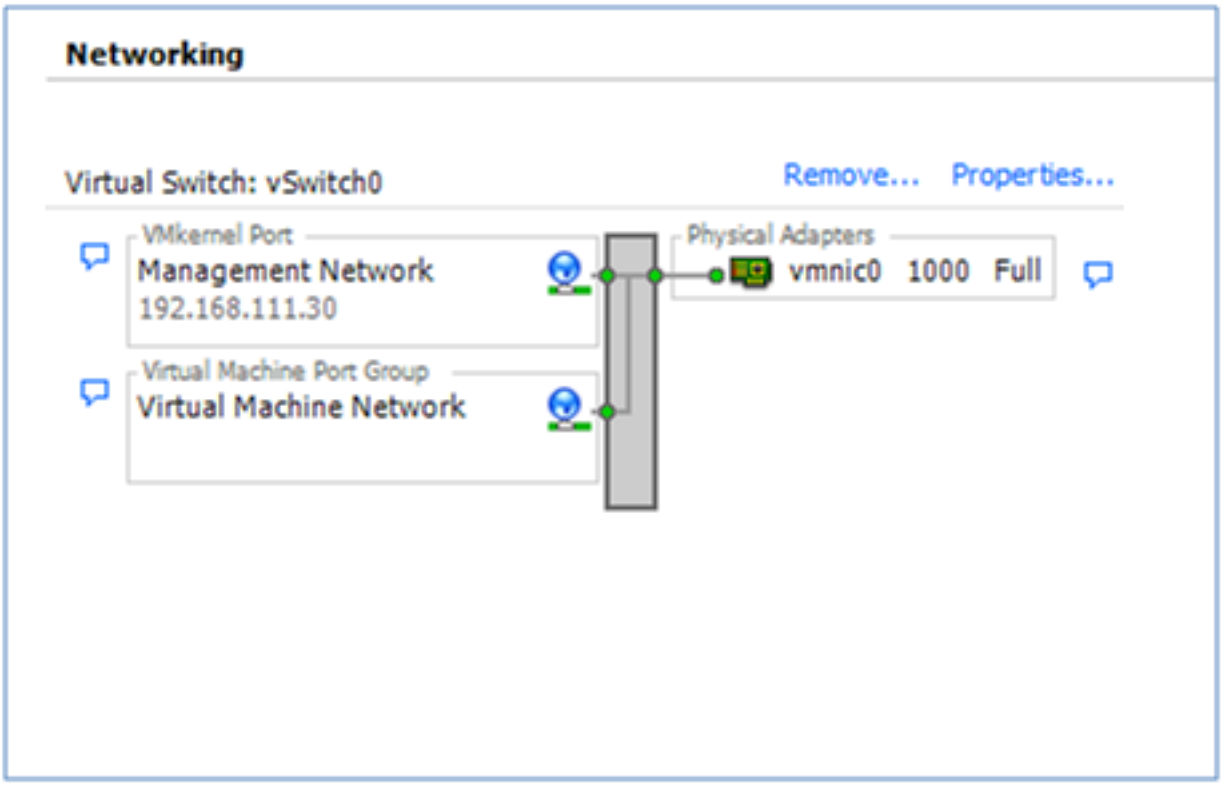
Congratulations, the installation of pfSense software on ESXi is complete!

34.71 Installing pfSense Software on vSphere 5.x using vSphere client

34.71.1 Basic vSphere Networking

About vmnics, vSwitches, management and virtual machine networks

In the vSphere client the network diagram for an ESXi host may be viewed by clicking **Networking** on the **Configuration** tab:



After ESXi was installed, before it was possible to connect to it with the vSphere client, a physical network adapter (a "vmnic" in the diagram) had to be nominated to be the ESXi Management Network. An IP address also had to be assigned to Management Network interface on the the ESXi host, either through DHCP or manually through the console.

The network diagram above shows that the Management Network was assigned to vmnic0 and it has an IP address of *192.168.111.30*. (192.168.111.0/24 is my home LAN. Others will most likely be different.) Whatever subnet was chosen, the VMkernel Port in the diagram is the Management Network and that's what the vSphere client is now talking to.

ESXi will name the first physical NIC it finds `vmnic0`. If vmnic0 is the management interface, ESXi will have automatically attached a virtual switch, `vSwitch0`, to that interface.

In addition to the VMkernel port, ESXi will also attach a Virtual Machine Port Group to the vSwitch. In the diagram above it's labeled as "Virtual Machine Network". The VM Port group is where Virtual Machines can be attached to this virtual network.

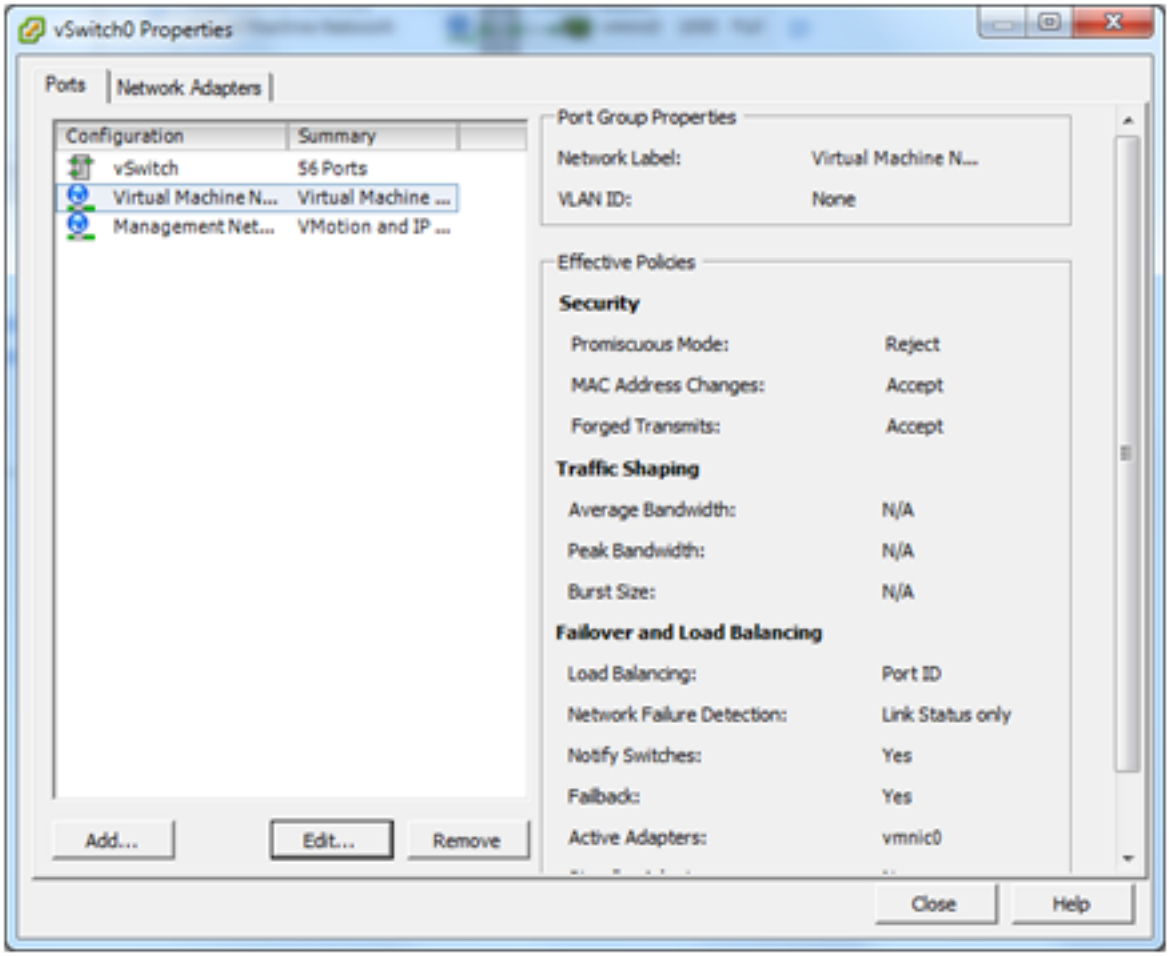
In summary, in the above diagram, vSwitch0 has both a VM Port Group (Virtual Machine Network) and a VMkernel Port (Management Network) attached.

Creating the LAN

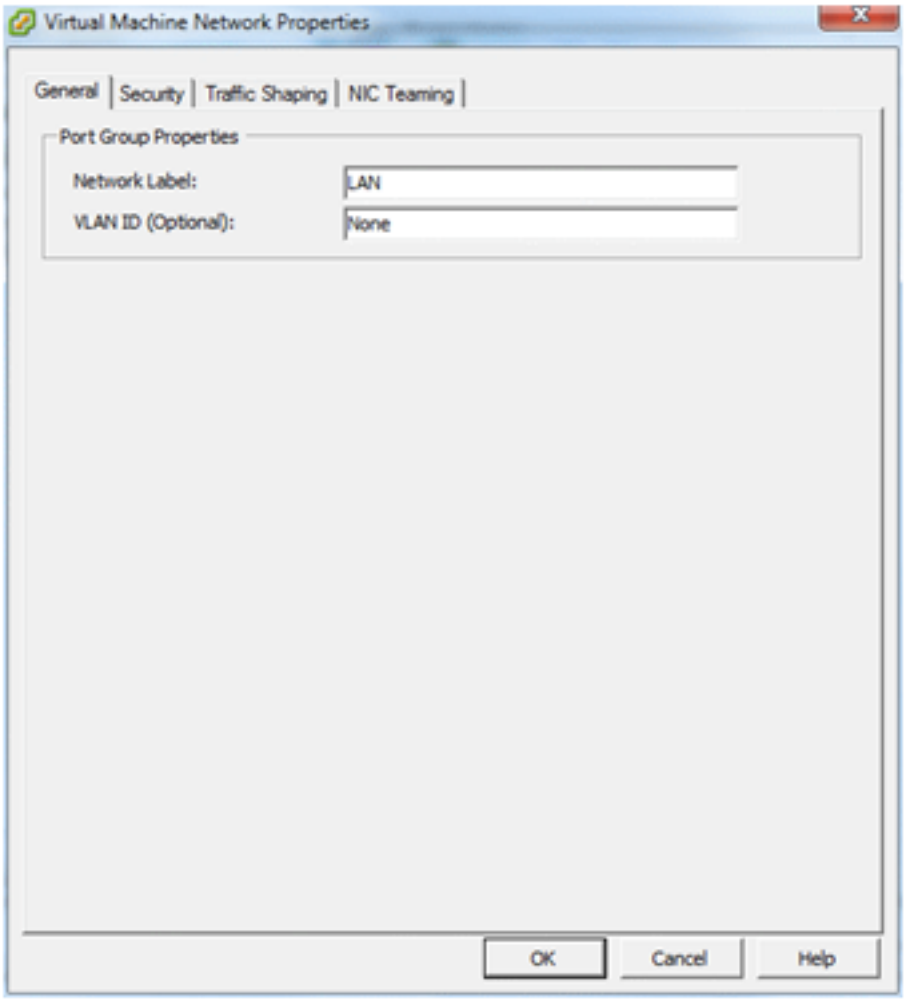
In a small network it is quite common to use the Virtual Machine Port Group on vSwitch0 to provide the LAN interface for the pfSense firewall. That allows access to the LAN side of the pfSense virtual machine and to manage the ESXi host with the vSphere client from a single PC. Of course, the virtual machine (e.g., the pfSense firewall) and the ESXi management interface must have different IP addresses.

*COMMENT: I must say here that I always separate the ESXi Management network from other networks. I won't go into the detail but there are some very good reasons for doing this. Without using VLANs, though, separation would mean that an additional NIC on the ESXi host would be dedicated only for ESXi management. What's more, another NIC would be required in the vSphere client PC to connect to the management NIC on the ESXi host. To follow that path and enough NICs are available, simply delete the Virtual Machine Port Group by clicking the Properties link above **vmnic0*, highlight the VM Port Group and click Remove.*

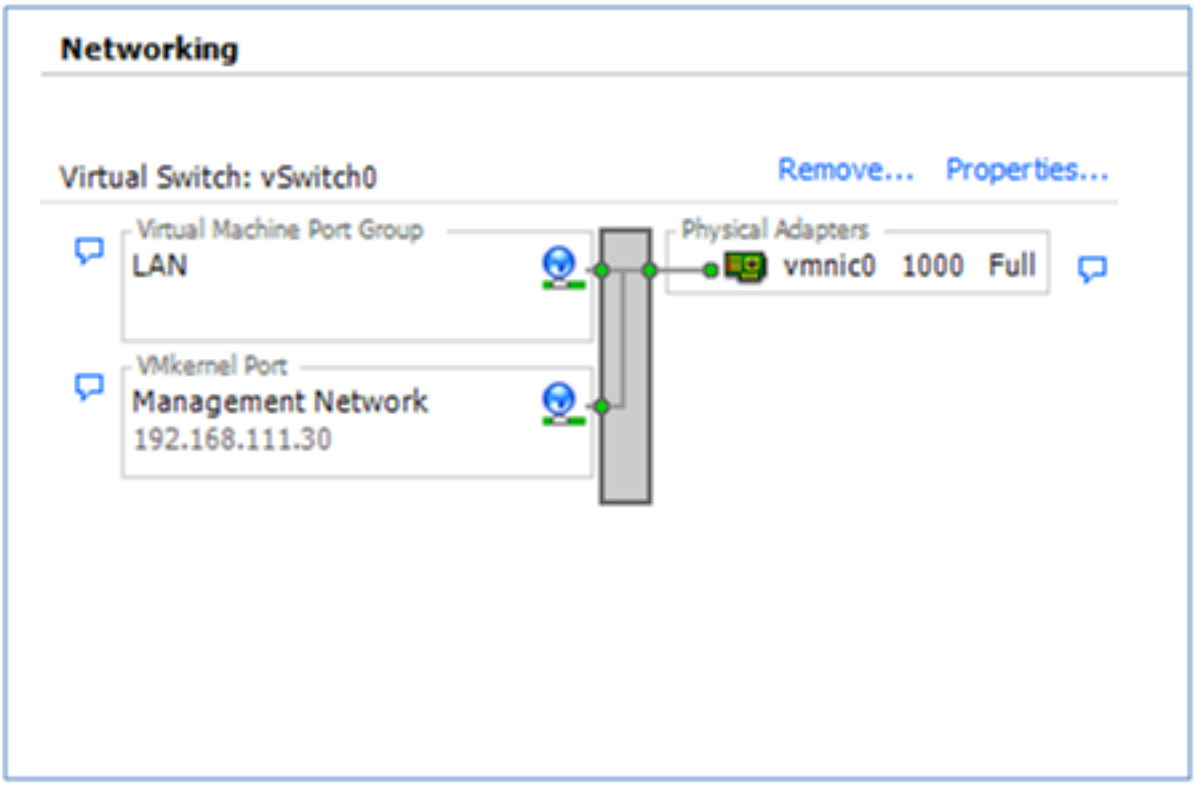
Assuming there are only two NICs in the ESXi host, rename the VM Port Group from "Virtual Machine Network" to something a bit more meaningful. Click the **Properties...** link for vmnic0:



Highlight the Virtual Machine Network and click the Edit button.



Change the **Network label** to "LAN" and click OK then Close.



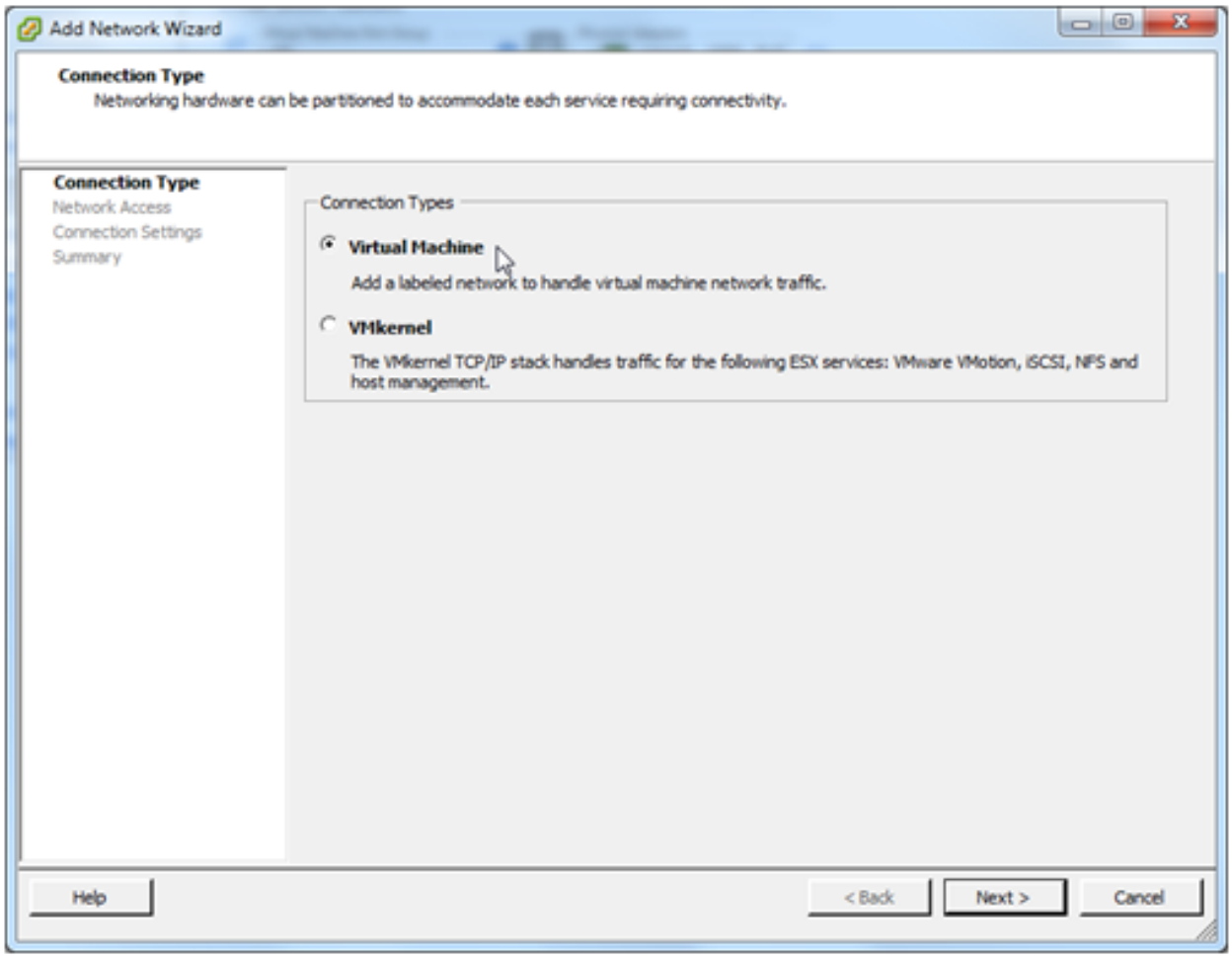
This makes life a little easier when we assign virtual network interfaces to the pfSense instance.

Creating the WAN

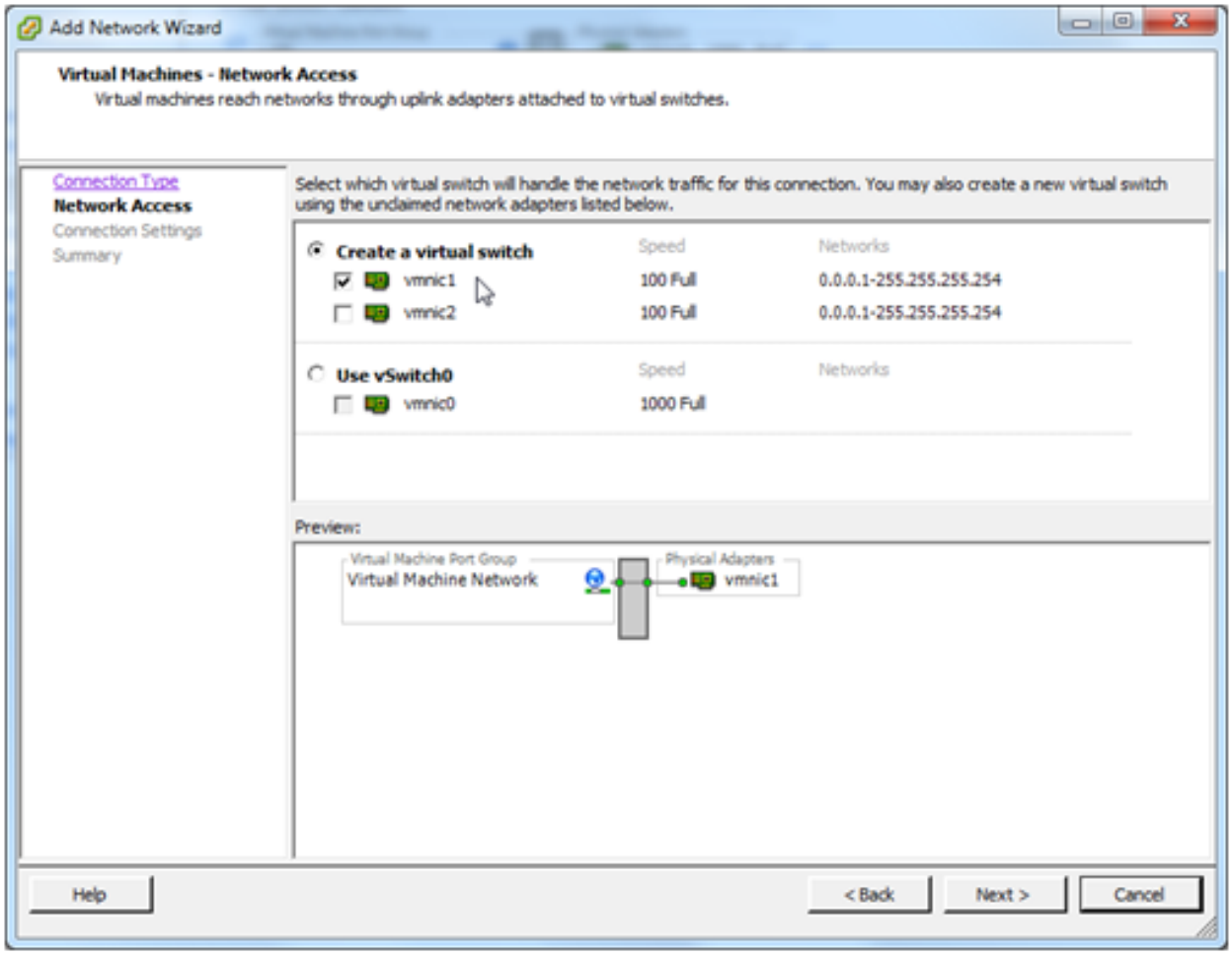
As we're not going to deal with VLANs here, a second physical NIC is required in the ESXi host. This will be the WAN interface.

HINT: If multiple physical interfaces are available in the ESXi host, it can be a bit of a struggle to work out which one has been identified as vmnic1, vmnic2 and so on. If the MAC address of each NIC is noted down along with the slot it occupied when it was installed in the machine, look at the Network Adapters screen under the Configuration tab to match up the MAC addresses (new to ESXi 5). However, having that foresight is rare, so lacking that information the easiest way to match physical NICs to vmnics is to plug a PC or switch into them, one at a time. The speed and duplex on the Networking or Network Adapters screens should change as the interface comes up. Because VMware didn't provide a Refresh link on the Network Adapters screen, refresh by navigating to somewhere else then going back.

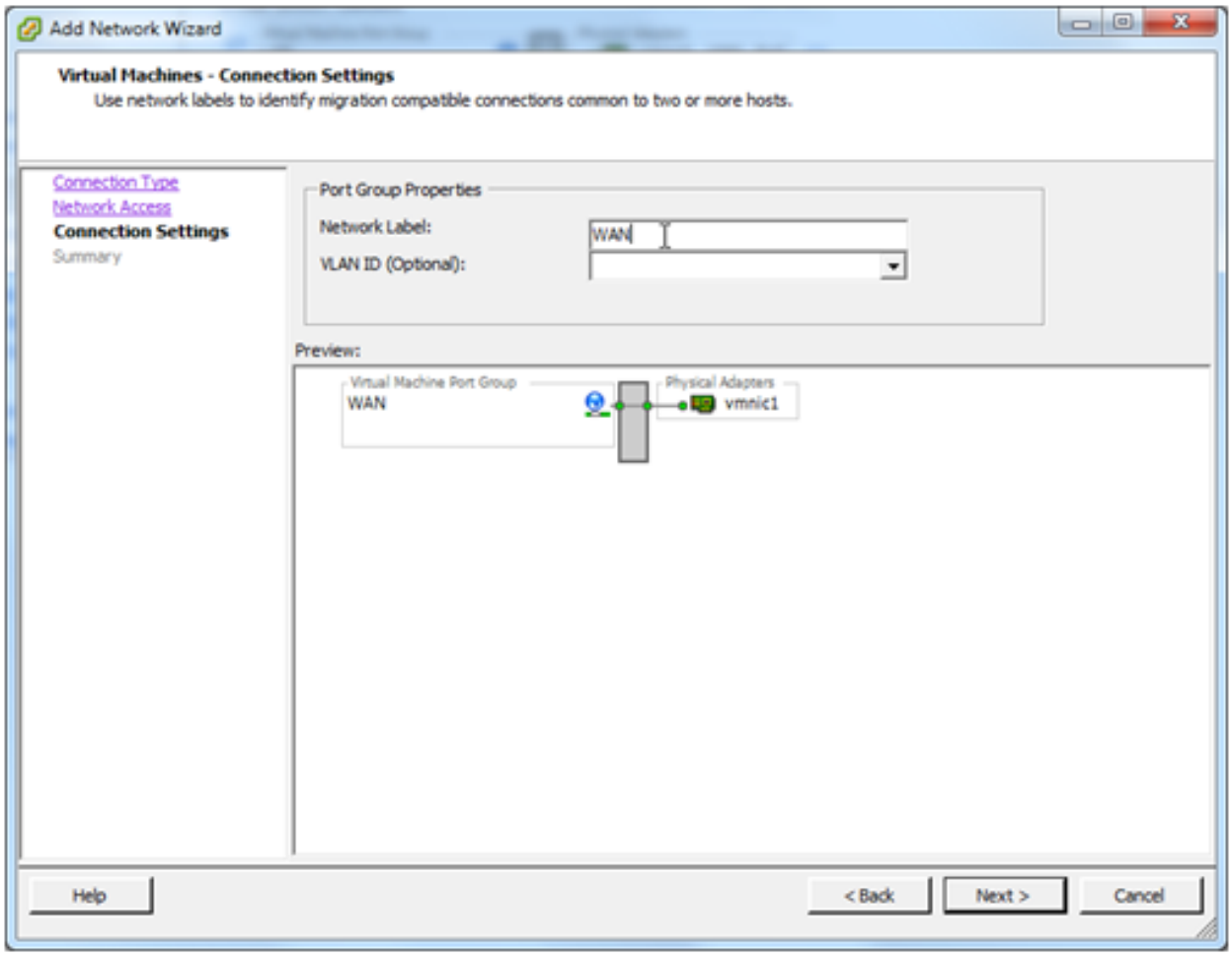
Now we need to link the second physical NIC (**vmnic1**), to a new vSwitch. Click the Add Networking link at the top right of the Networking screen and the following dialog will appear.



We are adding a Virtual Machine network so select that option and click Next.



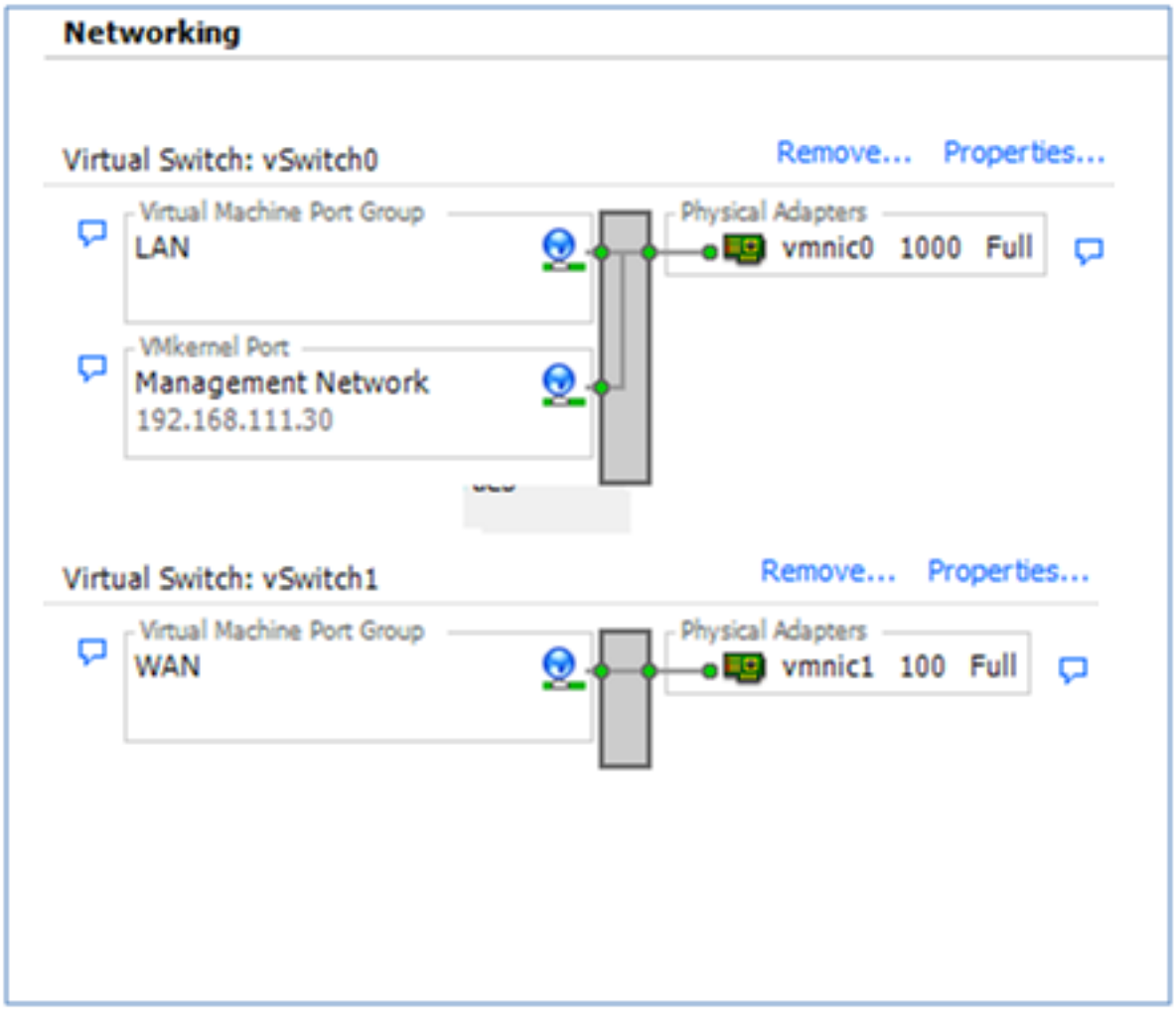
We want this NIC to be attached to a new vSwitch so select Create a virtual switch and check vmnic1. Click Next.



As we did with the LAN, let's give this VM Port Group a more meaningful name of "WAN". Click Next.

The next dialog simply confirms that everything looks OK. Click Finish.

The networking diagram should now look like this:

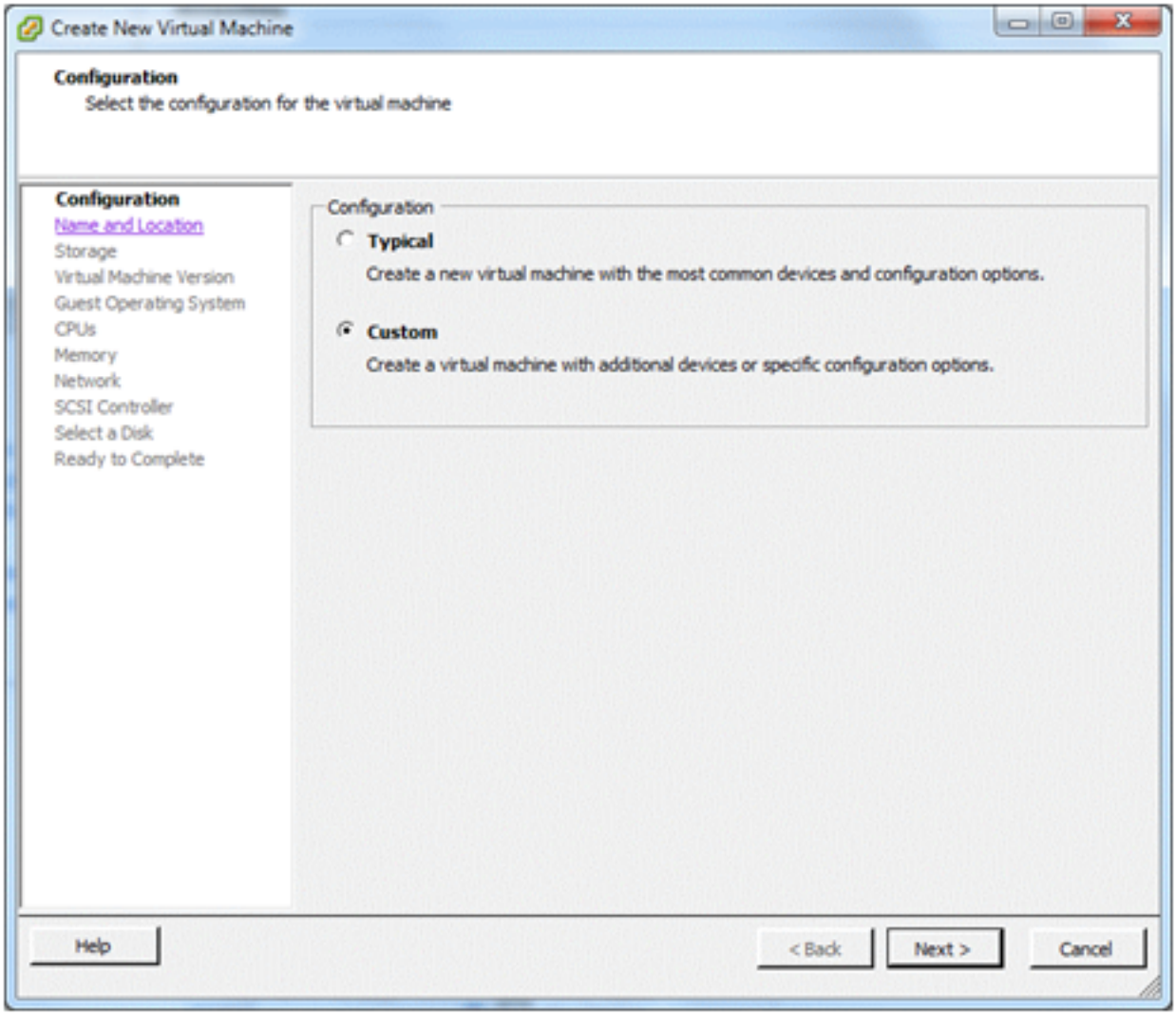


Now we can configure a new virtual machine on which pfSense software will be installed.

34.71.2 Configuring the Virtual Machine

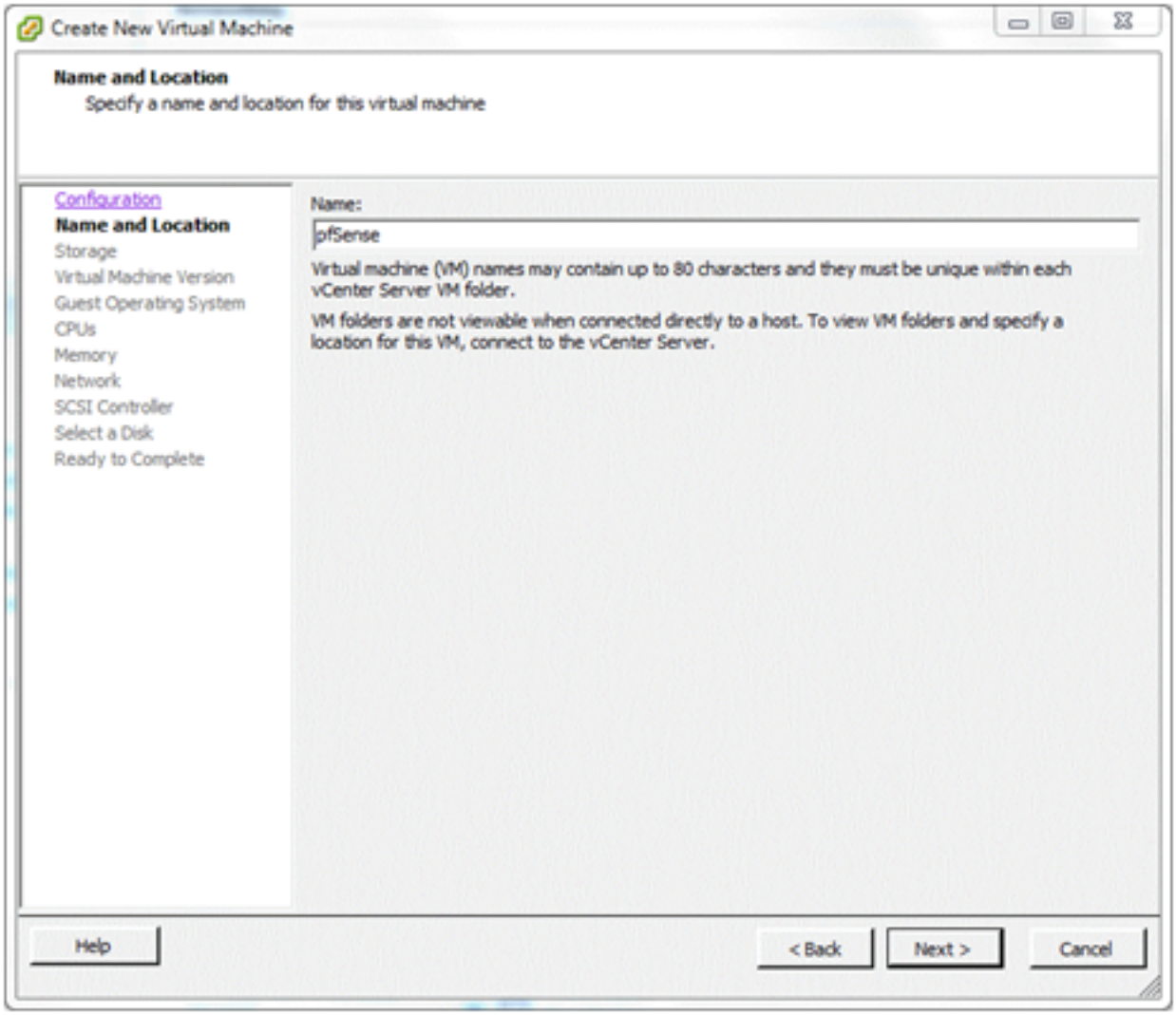
Right click the ESXi host in the left pane of the vSphere client and select **New virtual machine...**

Configuration



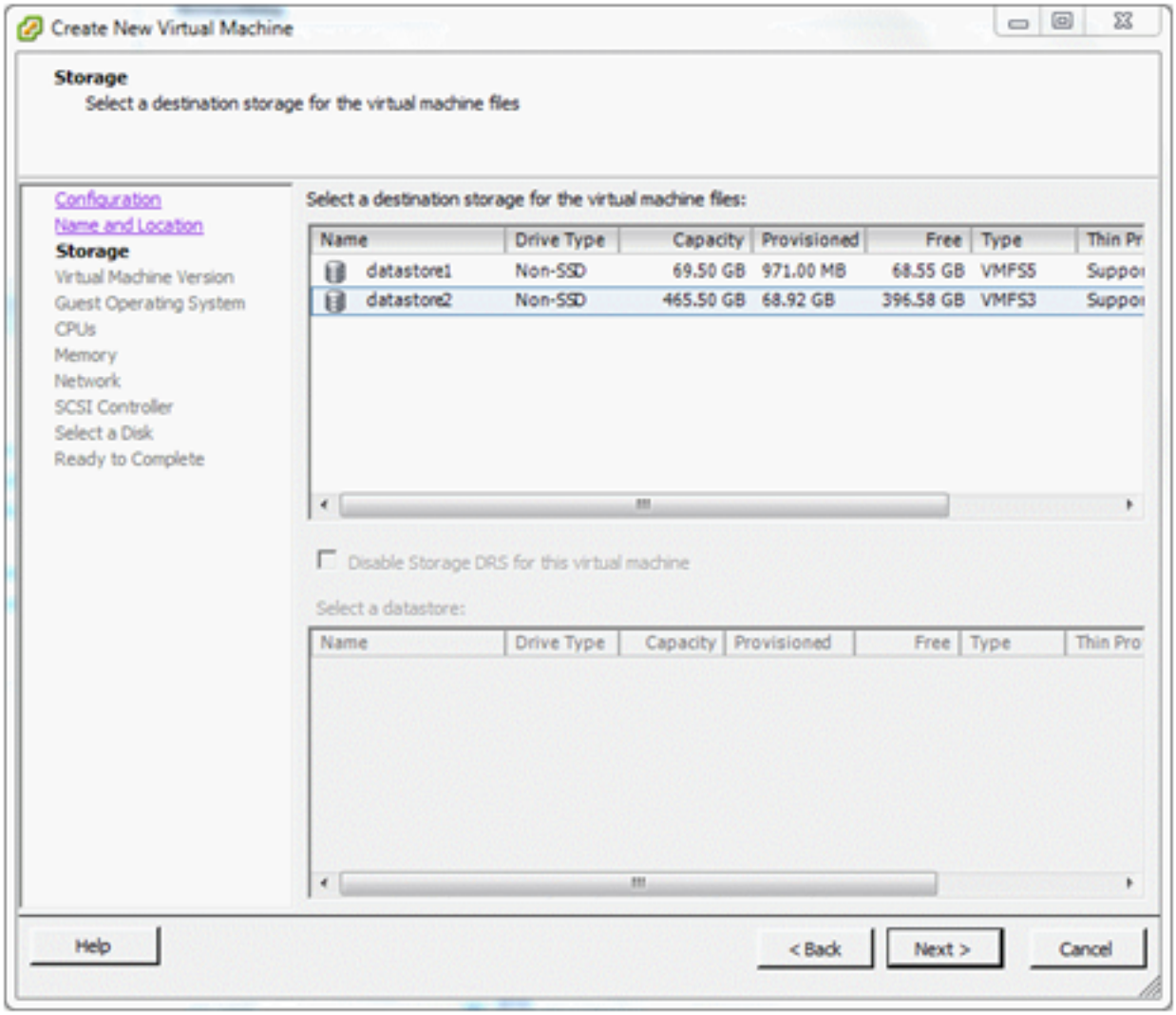
In the **Configuration** window, I always like to take the **Custom** option. (I've never really trusted what someone else thinks is "typical"). Click **Next**

Name and Location



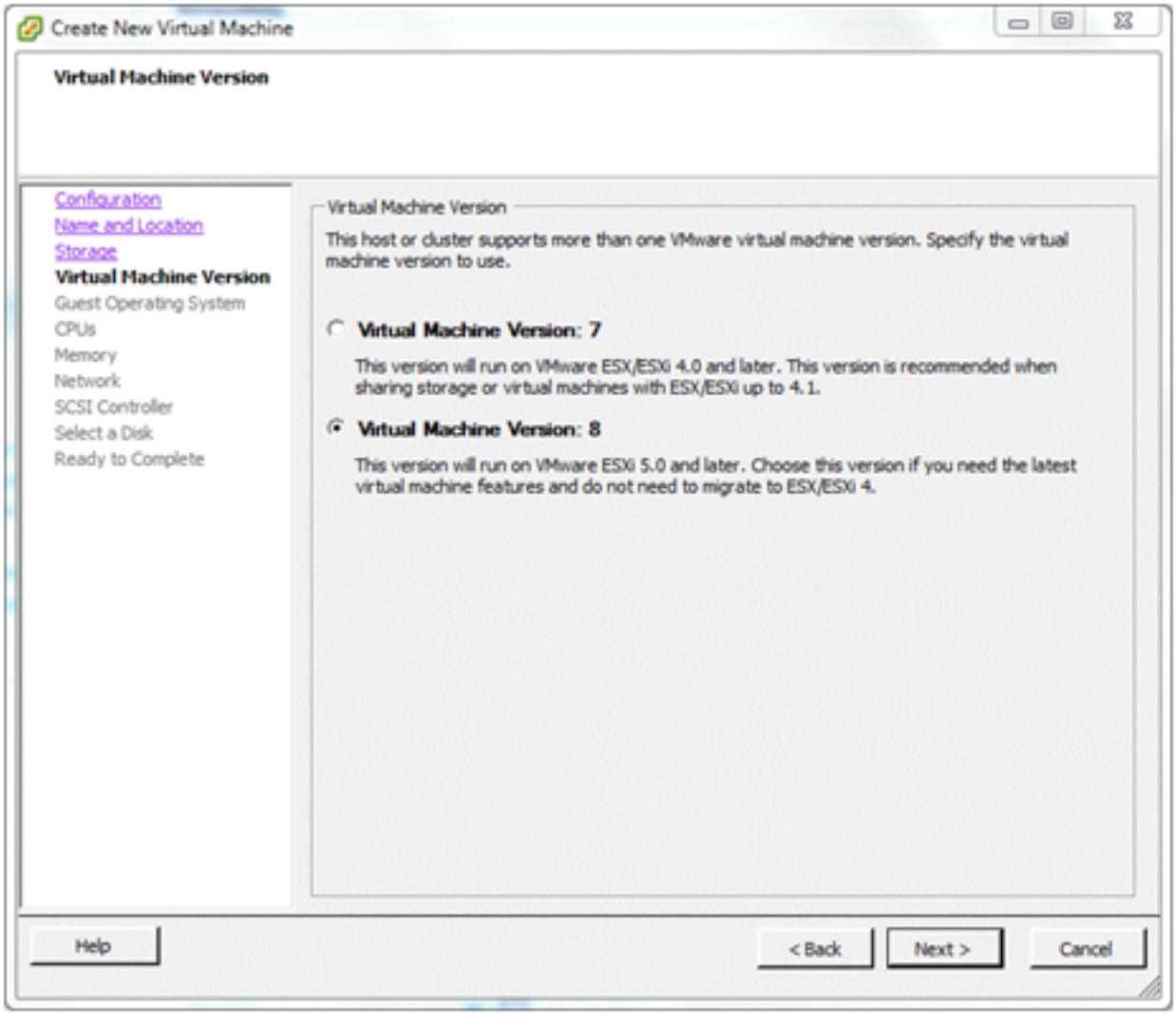
In the Name and Location window, let's give the virtual machine a meaningful name like "pfSense" and click **Next**

Storage



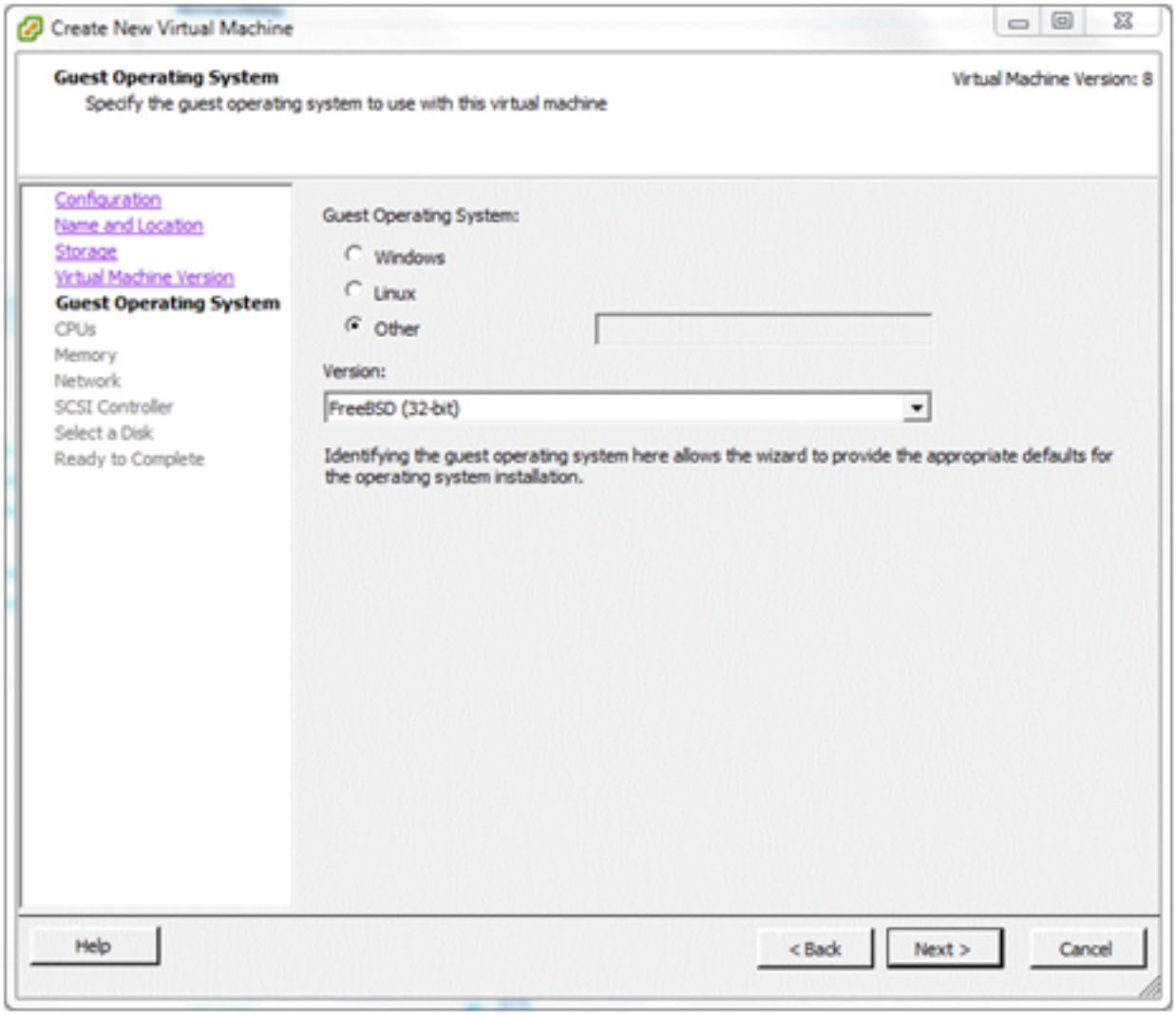
Now we need to decide where disk storage will be allocated to hold the configuration and operating files for the virtual machine. (This is not necessarily the same location as the file system for pfSense software, as shown later.) There are two datastores on this server – a small 80GB drive on which ESXi is installed and a 500GB disc which is for virtual machine storage. Highlight a datastore from the list and click **Next**.

Virtual Machine Version



Here is where the virtual machine version to use for the pfSense installation is configured in ESXi. Note the warning above. Select version 8 and click **Next**.

Guest operating System

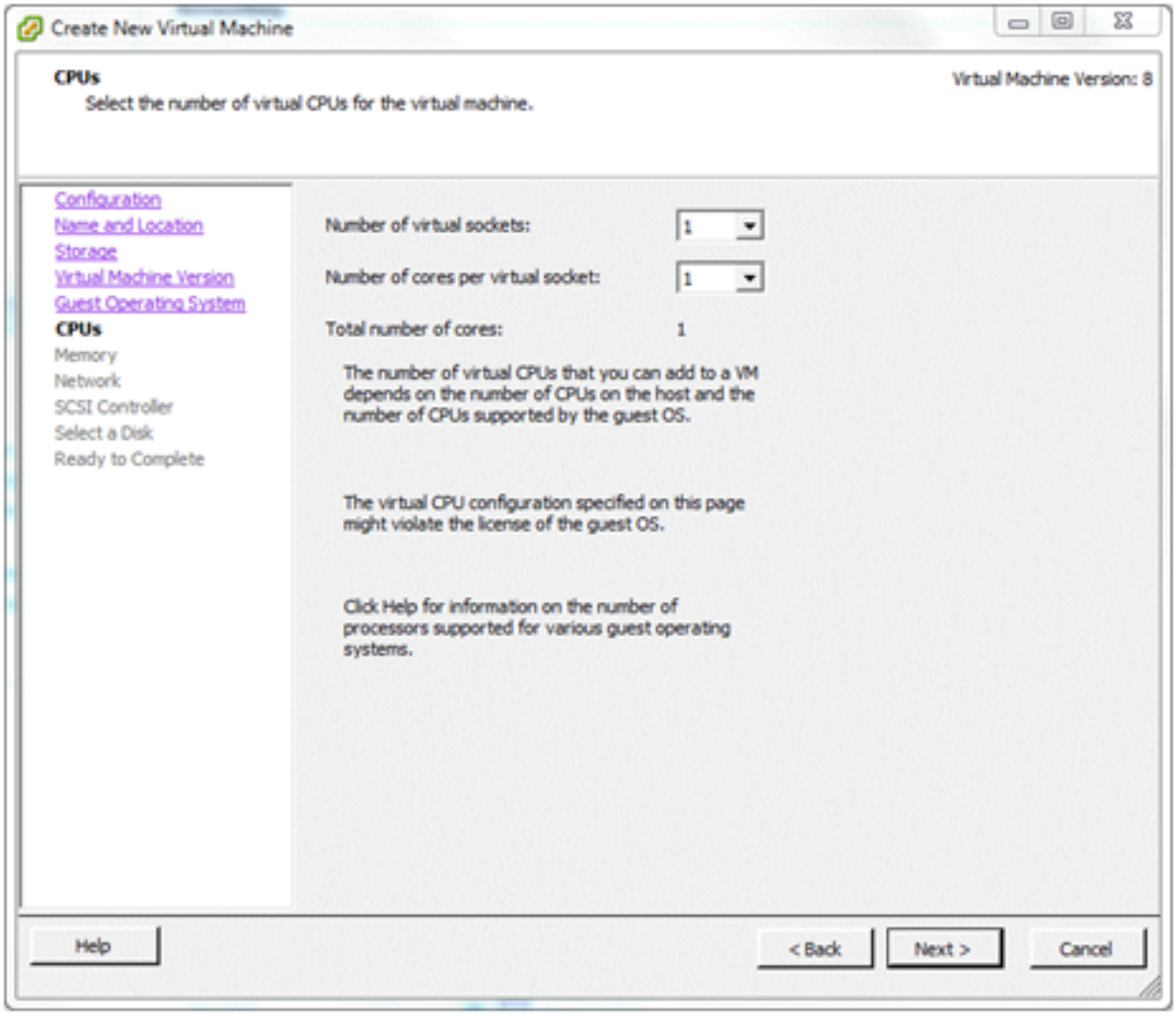


pfSense software is built on the FreeBSD operating system, not Linux. Select **Other** and chose **FreeBSD (32-bit)** or **FreeBSD (64-bit)**.

Make sure the hardware is capable of running 64-bit virtual machines, which it must be to run ESXi, and download the pfSense ISO image for installation, choosing the *amd64* architecture.

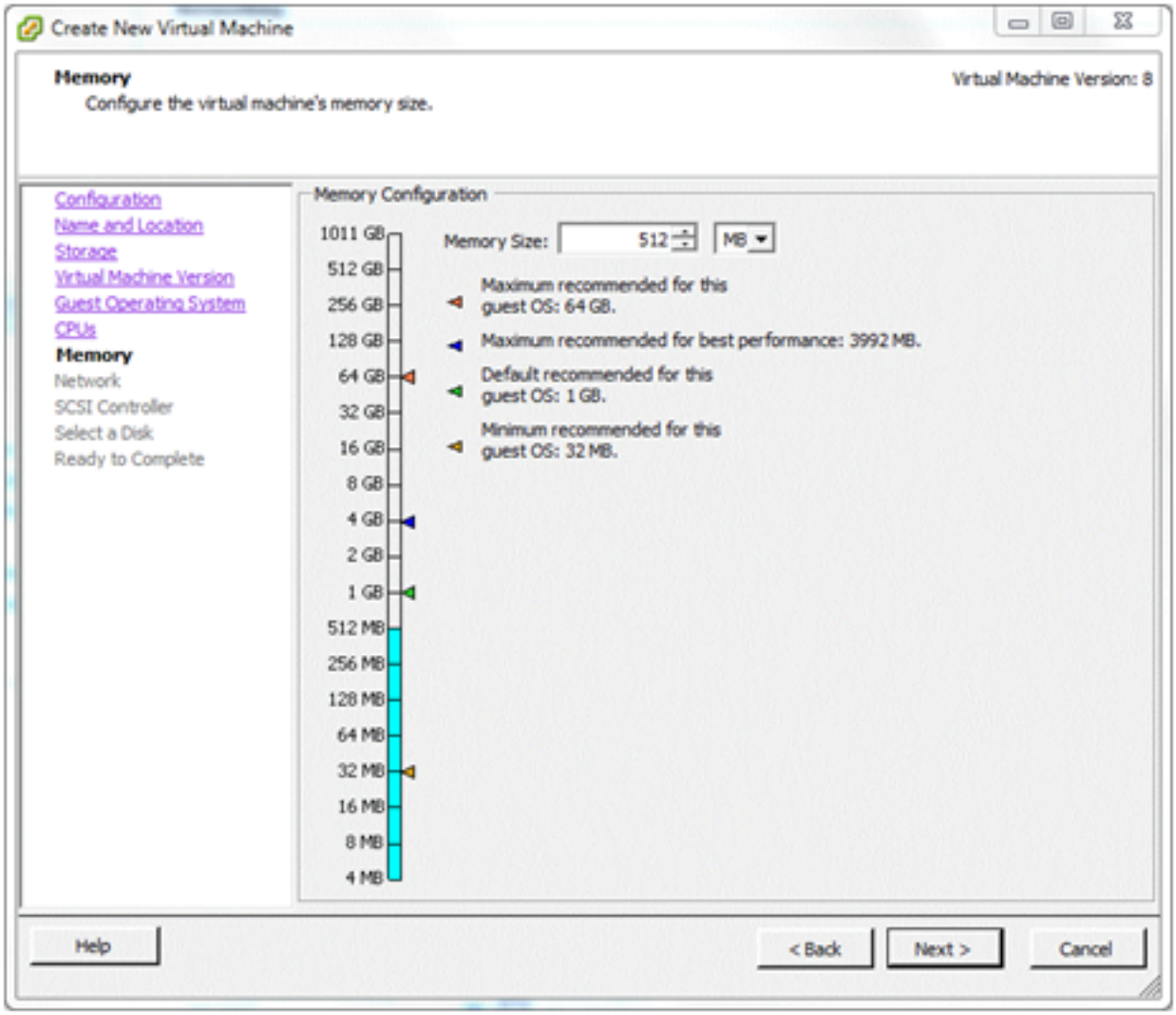
Click **Next**.

CPUs



To get started, a single-socket, single-core configuration will do for now. This and other virtual machine settings can always be changed later if needed. Click **Next**

Memory

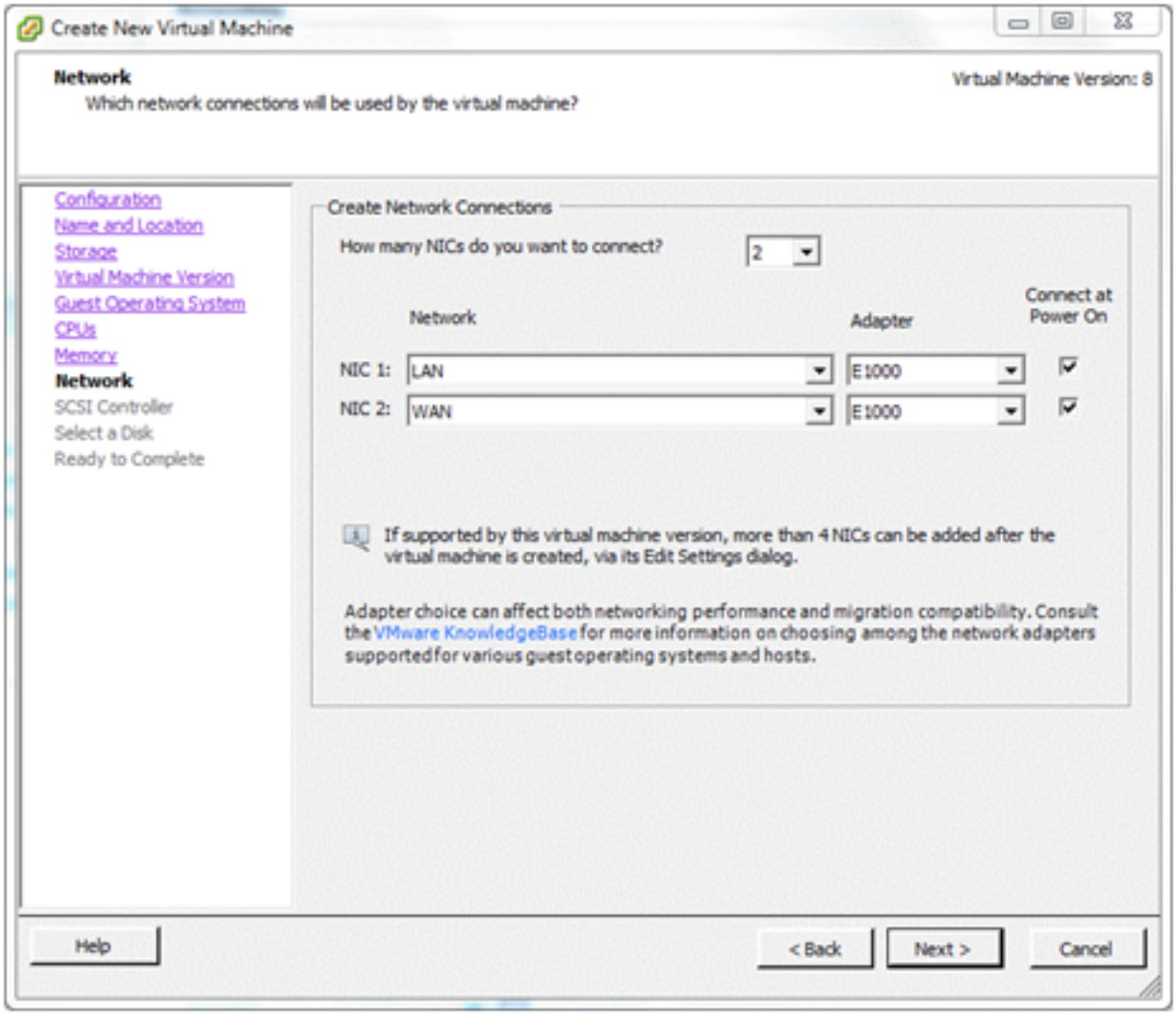


Depending on the number and type of packages that will be installed on the pfSense software, a basic pfSense VM should run comfortably in 512MB of RAM. A lot of simple, non-virtual installations run on old PCs with 256MB and less, so long as swap space is available on the disk. Given the low cost of RAM these days, allocating less than 512MB would not be advised.

If physical RAM on the ESX host is limited - perhaps because lots of other virtual machines will be running - the allocation on the pfSense VM could be reduced to, say, 384MB. If lots of memory-hungry packages will be run, give it more.

To change the memory allocation to one of the sizes shown on the scale of the memory "thermometer", click that value on the scale. Click **Next**

Network



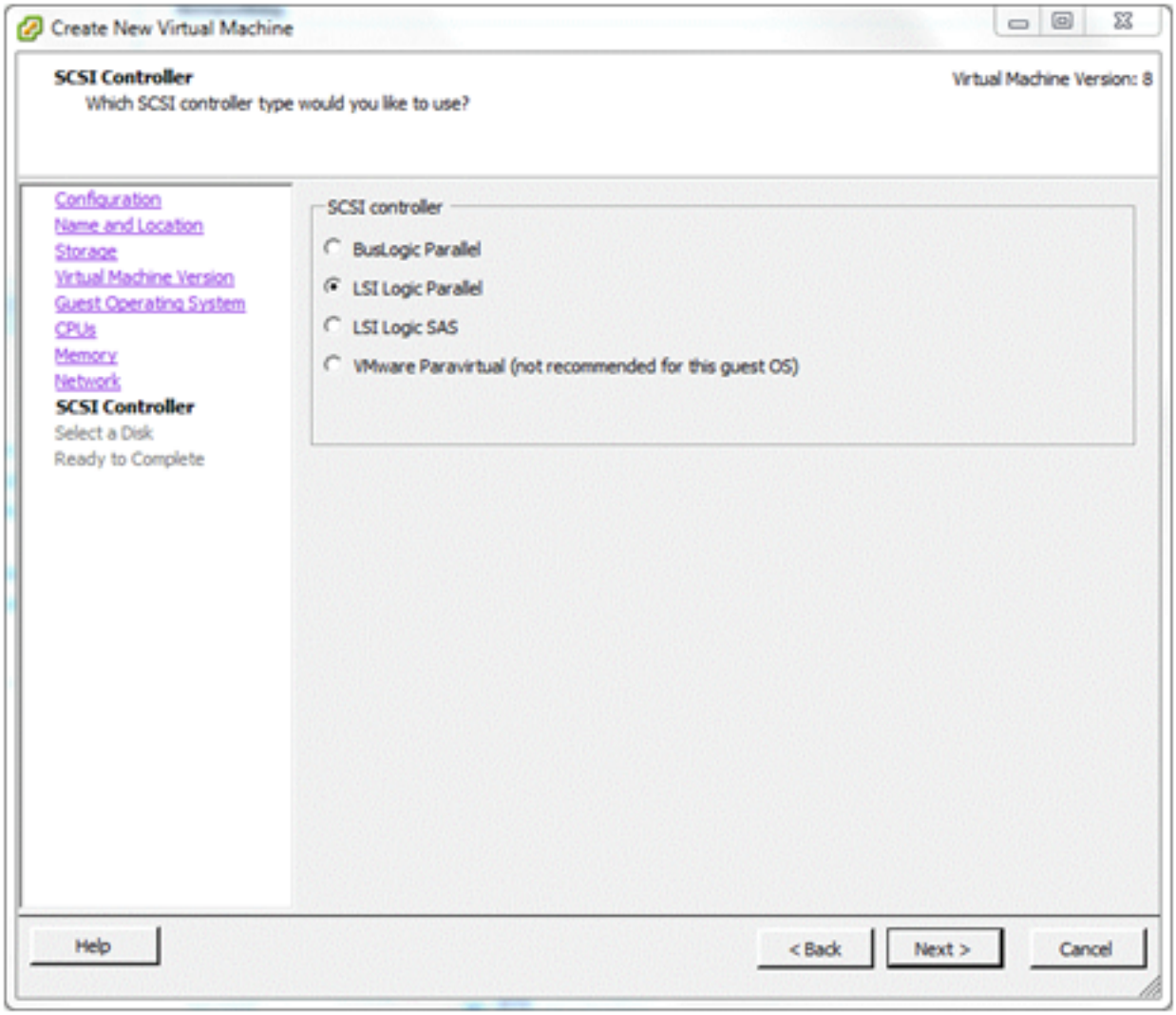
Remember that the two virtual networks were renamed to LAN and WAN. This is where we attach those networks to the pfSense virtual machine.

Select the number of virtual NICs for use by the pfSense VM. In this case it will be *2*. Now, using the drop-down lists assign **NIC 1** on the virtual machine to the WAN network. Assign NIC 2 to LAN. (This is why the virtual machine port groups were given these names – they are much easier to recognize.)

Note: On pfSense software version 2.2 and later, the choices in the default configuration are *em0* for WAN and *em1* for LAN, so WAN should be assigned to NIC 1. This may differ from the screenshots shown here. The interface assignment prompt will no longer appear for hosts using *em* NICs, so be careful not to attach a LAN to *em0*!

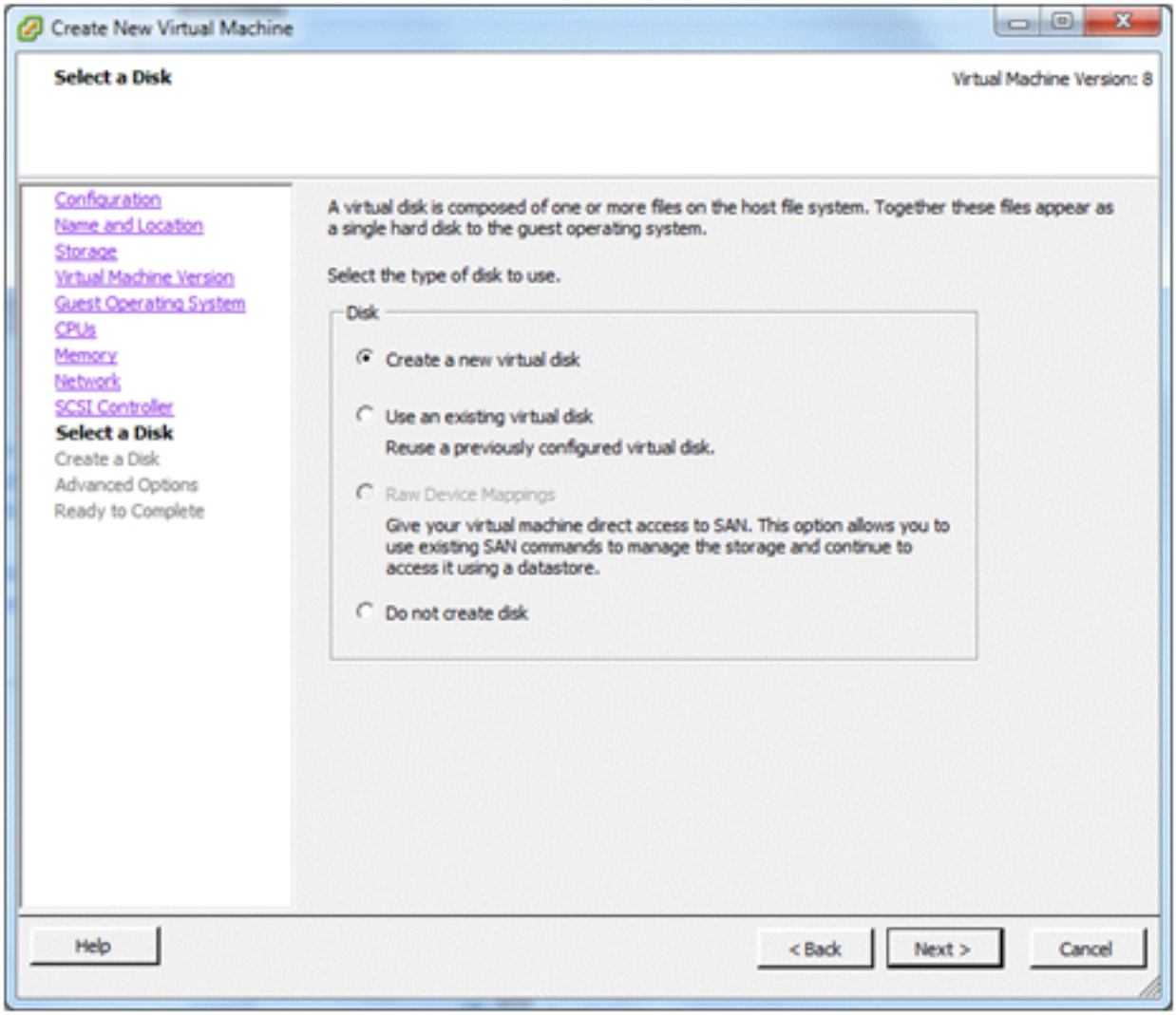
Note that for each NIC an **Adapter** type may also be selected. Different adapter types may give better or worse performance (and some may not work at all) but that is beyond the scope of this document. To get started, choose the dependable **E1000** type for each adapter. Make sure that **Connect at Power On** is checked and click **Next**.

SCSI Controller



An emulation of an LSI Logic SCSI controller is offered on this system and, as far as I know, the recommendation is based on the operating system of the virtual machine intend to be installed. Accept the default and click **Next**.

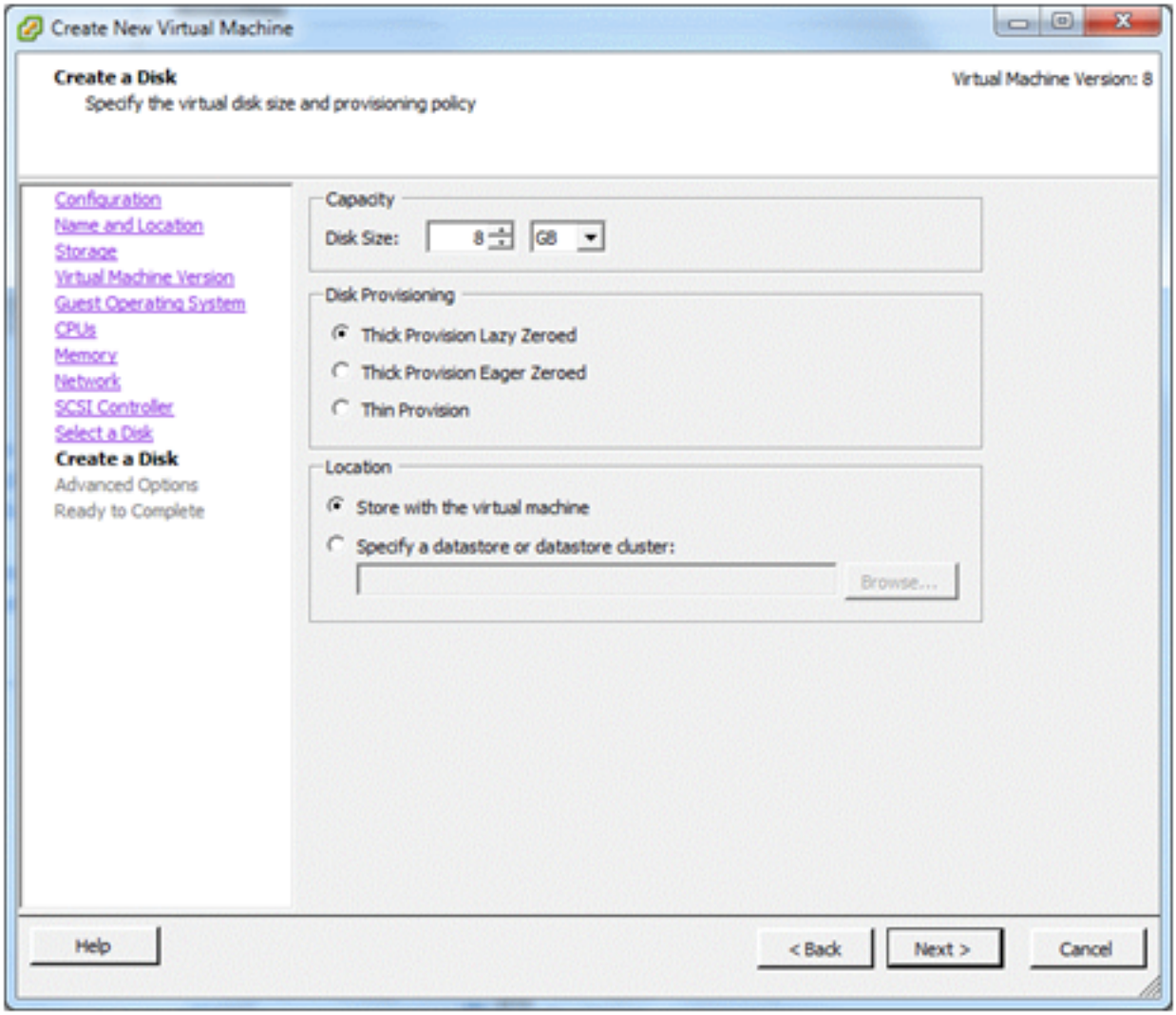
Select a Disk



This is where the operating system will build its file system.

Choose **Create a new virtual disk** and click **Next**.

Create a Disk

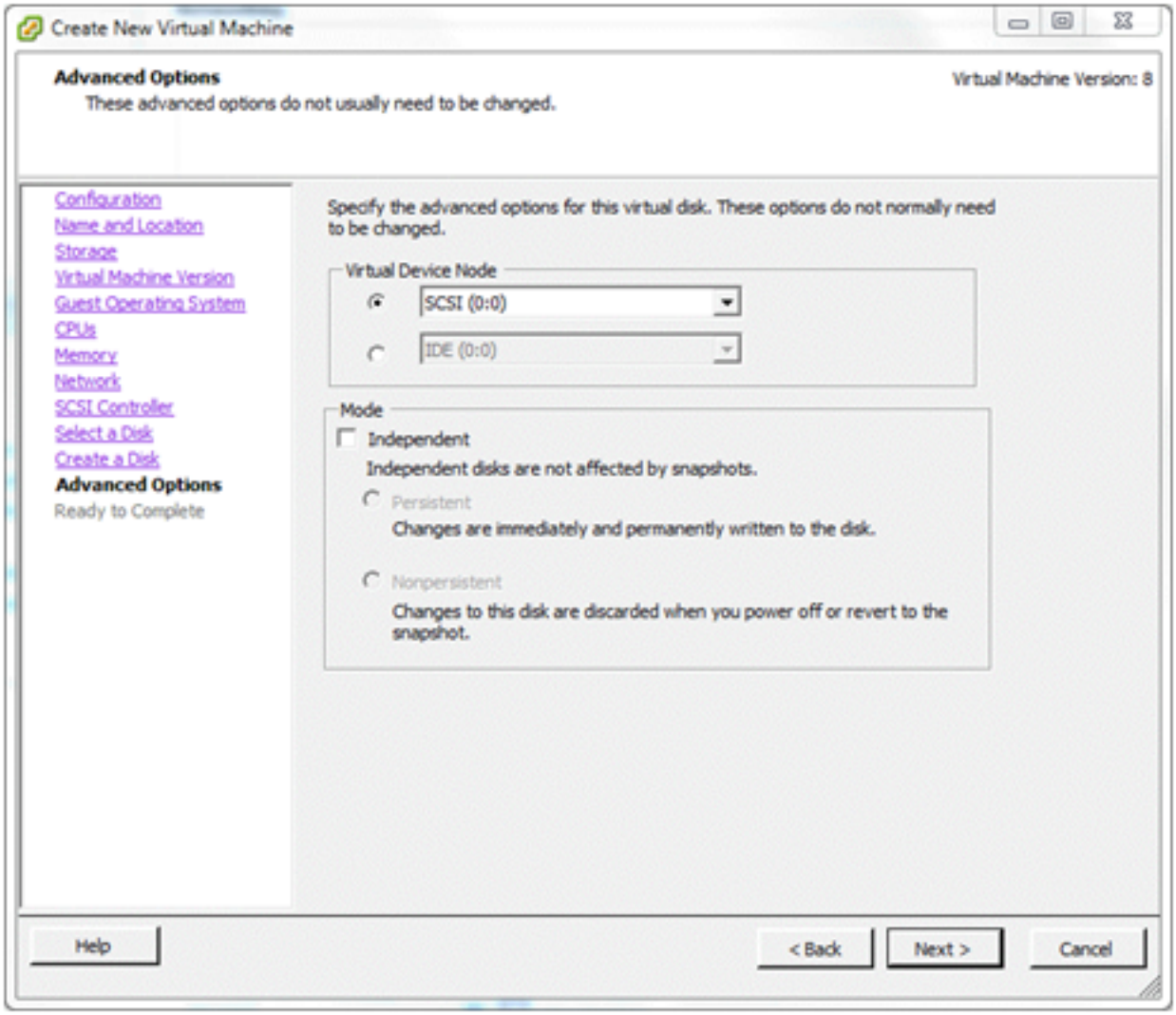


In this example, the virtual disk was given a capacity of *8GB* but there is quite a lot to spare and *8GB* isn't really that much these days.

Under **Location,** keep the virtual machine's hard disk with the virtual machine itself. Read the help to learn more about this set of options, if desired.

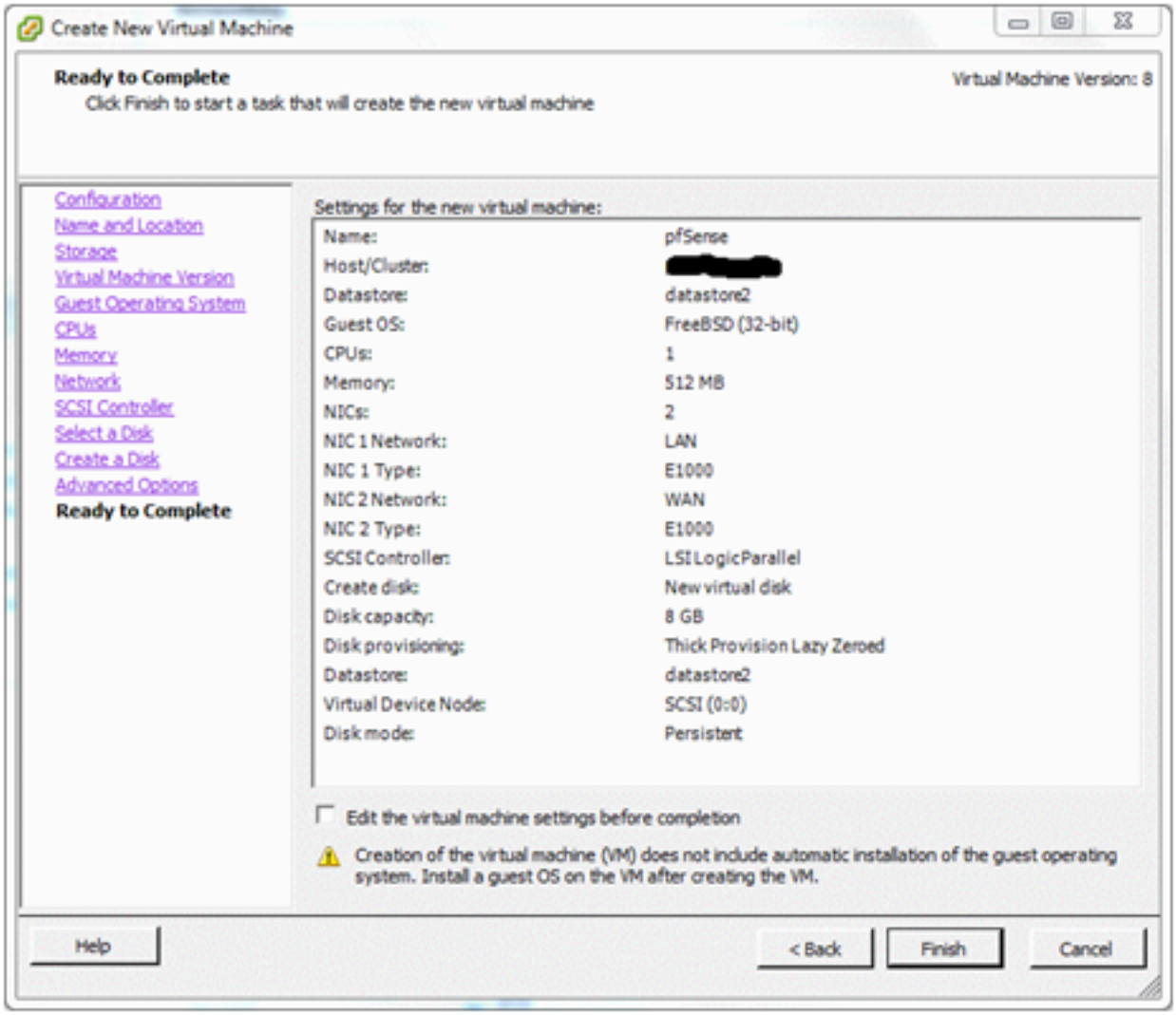
Click **Next**.

Advanced Options



Like it says – these options do not normally need to be changed. **Next.**

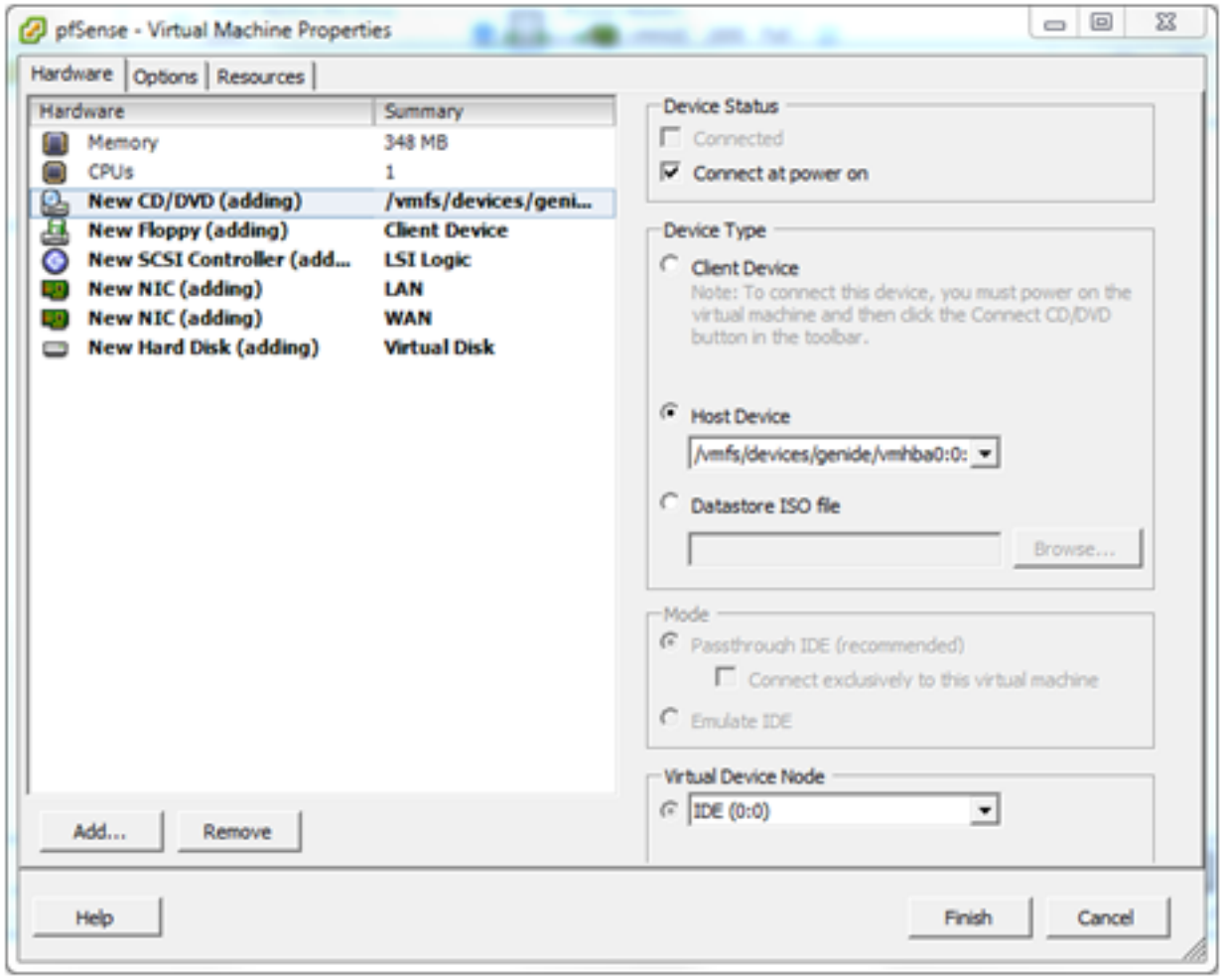
Ready to Complete



Now a summary of what has been configured so far for this virtual machine is displayed.

Before finishing, check the box **Edit virtual machine settings before completion**. The label on the **Finish** button will change to **Continue**. This will allow the boot CD from which the pfSense software will be installed to be configured. Click **Continue.**

Editing the Virtual Machine's Properties



In the Virtual Machine Properties dialog select the **Hardware** tab and then the line **New CD/DVD (Adding)** line. In the right-hand pane choose the location of the CD/DVD drive:

- **Host Device**

If a CD/DVD drive is available in the ESXi host, select the CD/DVD drive and check **Connect at power on**. This change allows pfSense CD/DVD to be inserted into the host's drive and start installing a soon as the virtual machine is powered on.

- **Client Device**

To install from the CD/DVD drive in the vSphere Client PC, select the Client Device option. As the contents of the CD will be read across the network, this will be a bit slower than using a drive in the ESXi host. In addition, Connect at power on is not available.

- **Datastore ISO**

To install from an ISO image stored in an ESXi datastore, that is also an option but it won't be covered here. This is much faster than the other options, and more convenient to keep the install media around for re-use.

Click **Finish**.

34.71.3 Installing pfSense Software

Booting the VM from CD/DVD

Option 1: Installing from the CD/DVD drive on the ESXi host

If the Host Device option was chosen in the Virtual Machine Properties above, slip the pfSense CD into the drive on the ESXi host.

In the left-hand pane of the vSphere client window, right-click the new pfSense virtual machine. A number of actions for VM are displayed, including **Power > Power on**. Select that or highlight the virtual machine and click the green arrowhead in the toolbar.

Now click the **Console** tab and the virtual machine will begin booting from the CD.

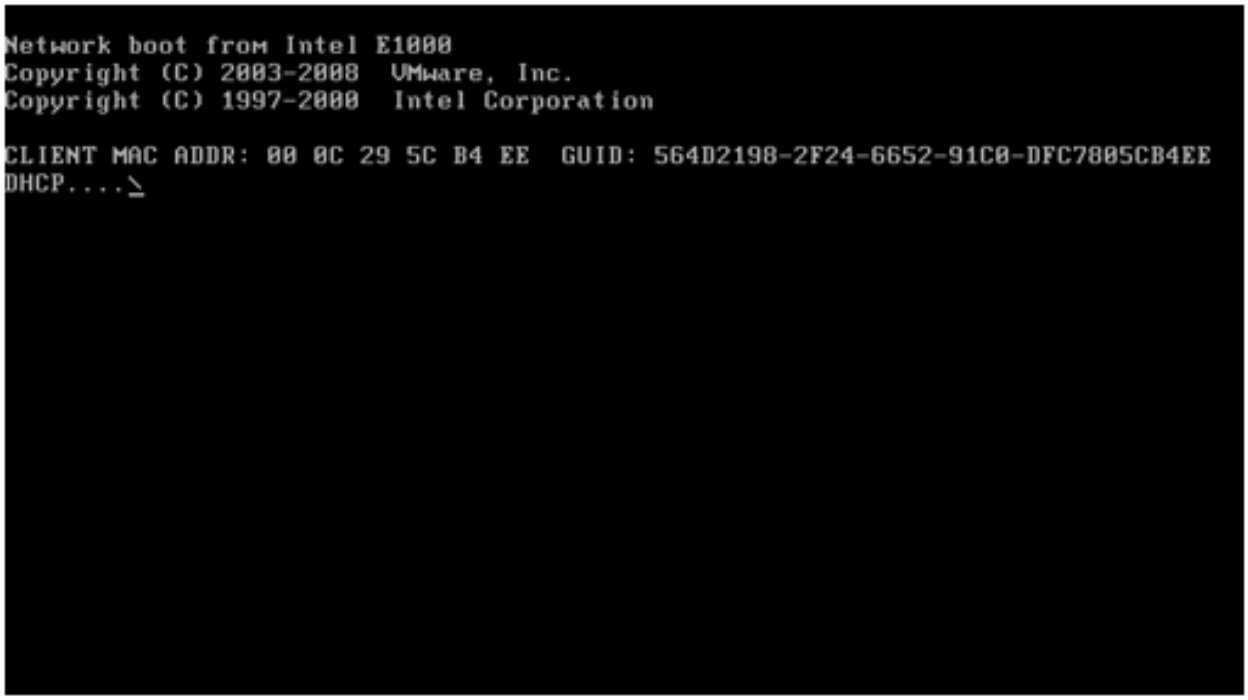
Skip to **Installing pfSense Software.**

Option 2: Installing from the CD/DVD drive on the client

If the option was chosen to use the drive in the client PC, put the CD into its drive. Remember that **Connect at power on** was not a choice if using the client's CD/DVD drive, so a little bit of extra work is needed to connect it after powering on the virtual machine.

In the left-hand pane of the vSphere client window, right-click the new pfSense virtual machine. A number of actions for VM are displayed, including **Power > Power on**. Select that or highlight the virtual machine and click the green arrowhead in the toolbar.

Now, with the virtual machine highlighted, click the **Console** tab.



Because the CD drive is not attached to the virtual machine yet, it may attempt to boot from the network or it may be showing an *Operating system not found* or some other error. Don't worry about this.

At this point (and only after the virtual machine has been powered on) the virtual machine may be attached to the CD/DVD drive on the client PC. Click on the toolbar icon that looks like a CD with a wrench/spanner. **CD/DVD Drive 1** will be offered in the menu and the available choices are displayed. Select Connect to D: (or whatever drive letter represents the CD/DVD drive on the client PC).

Right click the virtual machine in the left pane of the vSphere client and select **Guest > Send Ctl+Alt+Del**. This will reboot the virtual machine without disconnecting the CD/DVD drive.

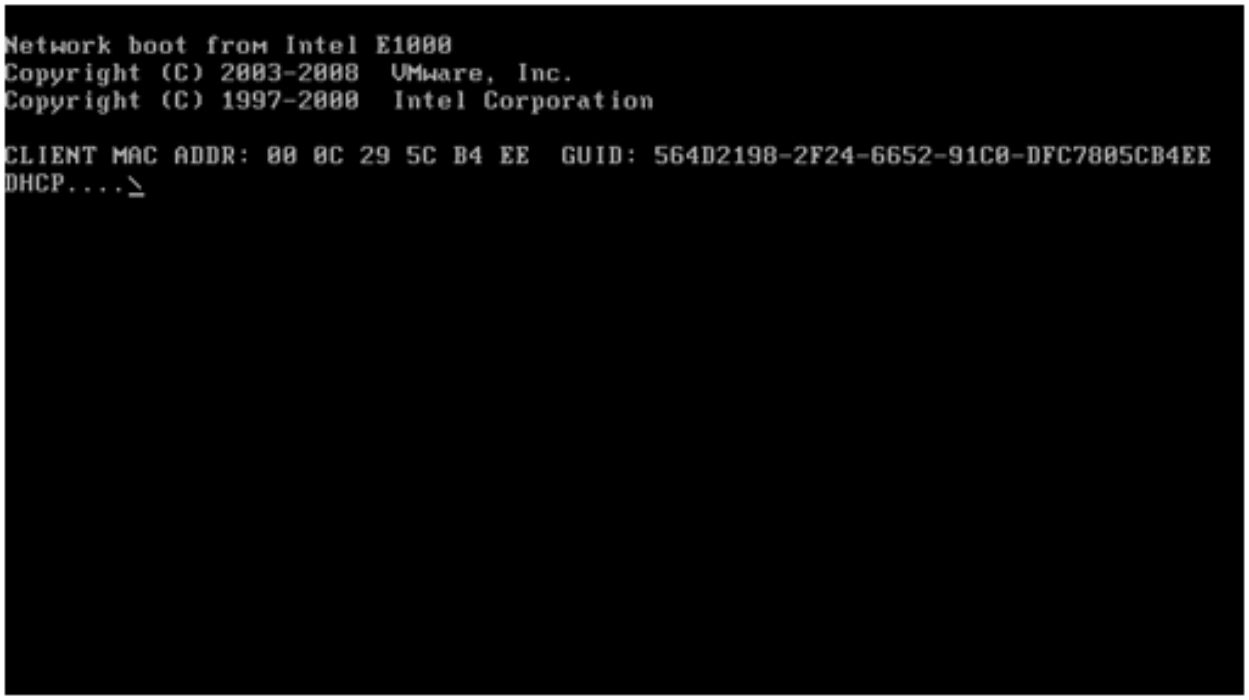
In the Console tab, the pfSense installer can now be seen booting from the CD.

Installing pfSense Software

If everything has gone well the pfSense boot menu be shown.

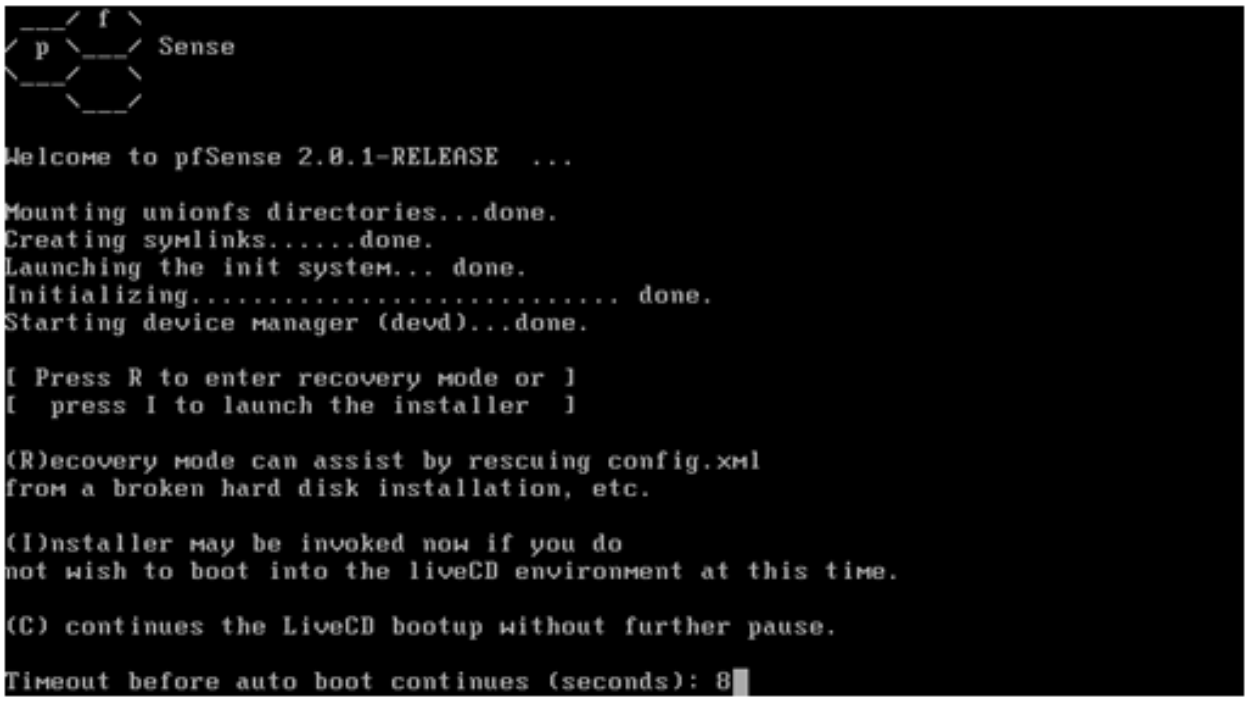
What follows is very much a standard pfSense installation procedure. However, it's included here to save jumping around between documents.

Note: To enter information through the virtual machine's console, it is necessary to click inside the console window. To release the cursor, press Ctl+Alt.



Allow the timer to expire and boot the pfSense software from the ISO image.

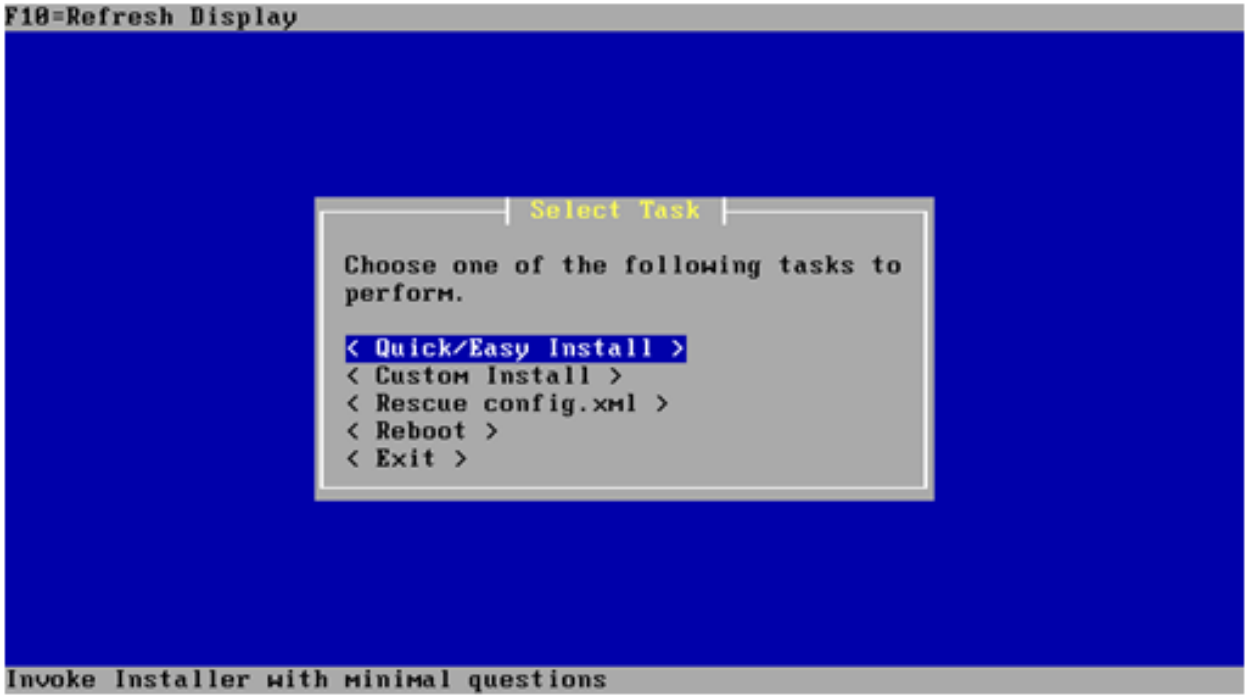
When the following console message is seen:

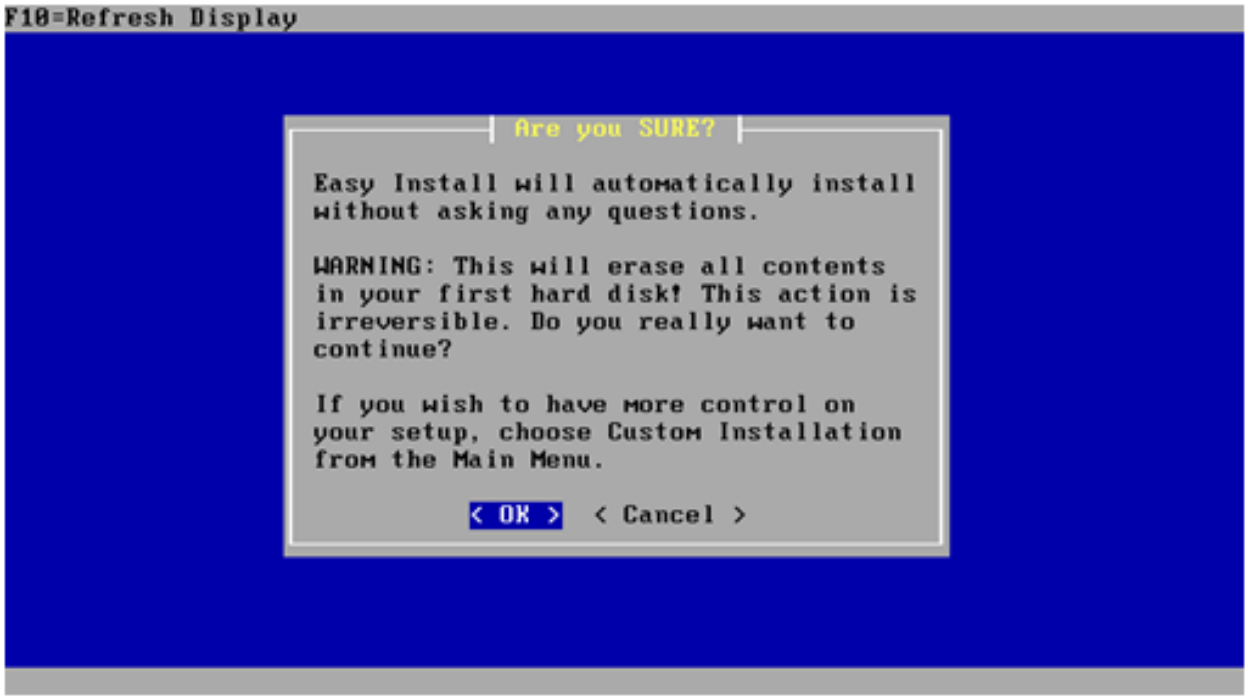


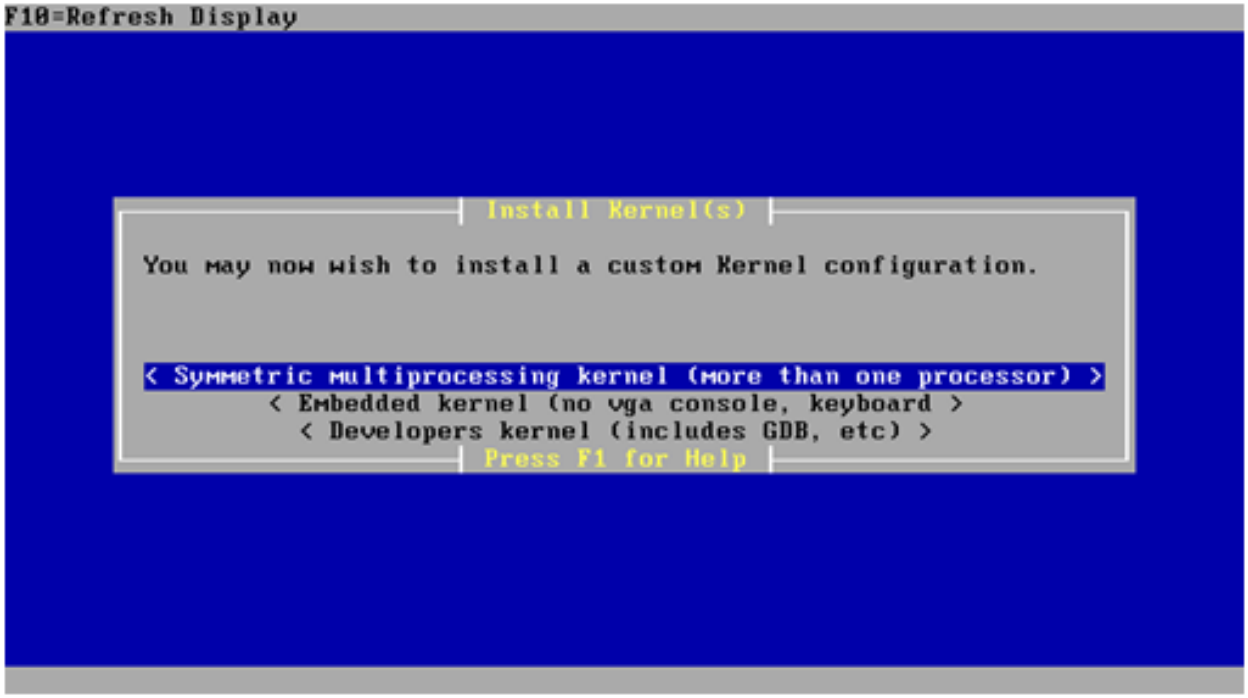
Type *i* to launch the pfSense installer.

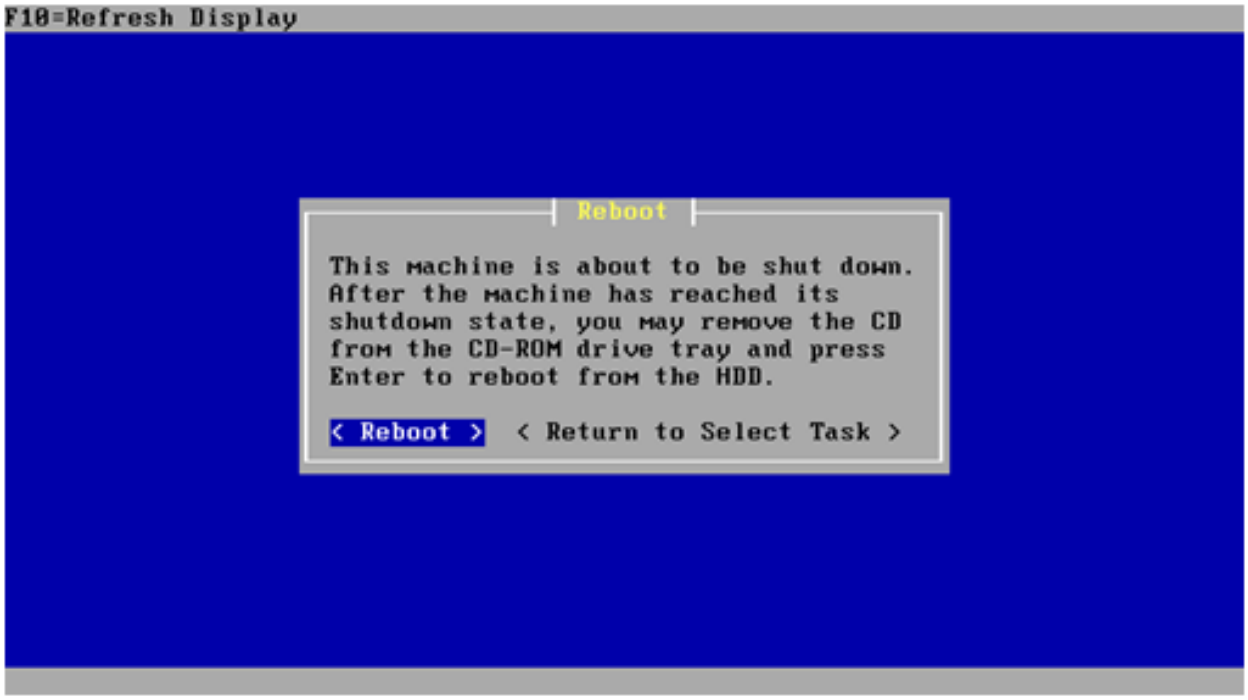
The next few screens are the standard pfSense install screens and are fairly self explanatory. Take the highlighted choice in each of the following screenshots:




F10=Refresh Display
Select Task
Choose one of the following tasks to perform.
< Quick/Easy Install >
< Custom Install >
< Rescue config.xml >
< Reboot >
< Exit >
Invoke Installer with minimal questions


F10=Refresh Display
Are you SURE?
Easy Install will automatically install without asking any questions.
WARNING: This will erase all contents in your first hard disk! This action is irreversible. Do you really want to continue?
If you wish to have more control on your setup, choose Custom Installation from the Main Menu.
< OK >
< Cancel >

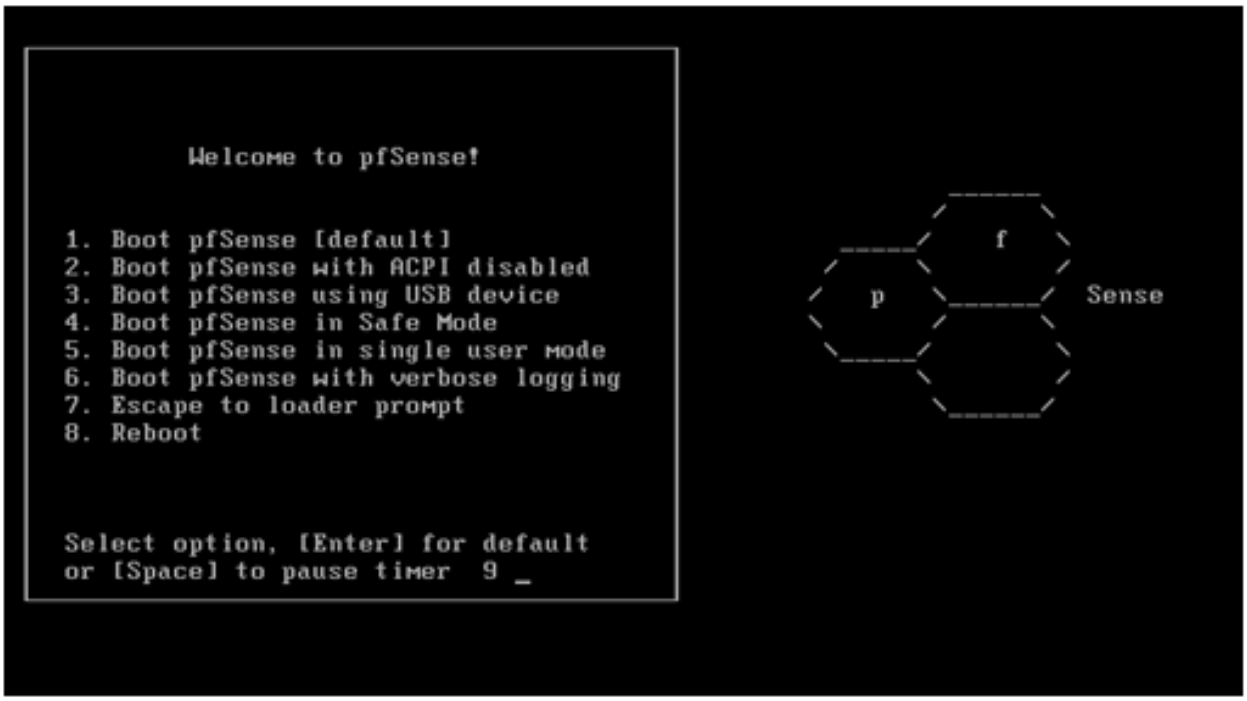




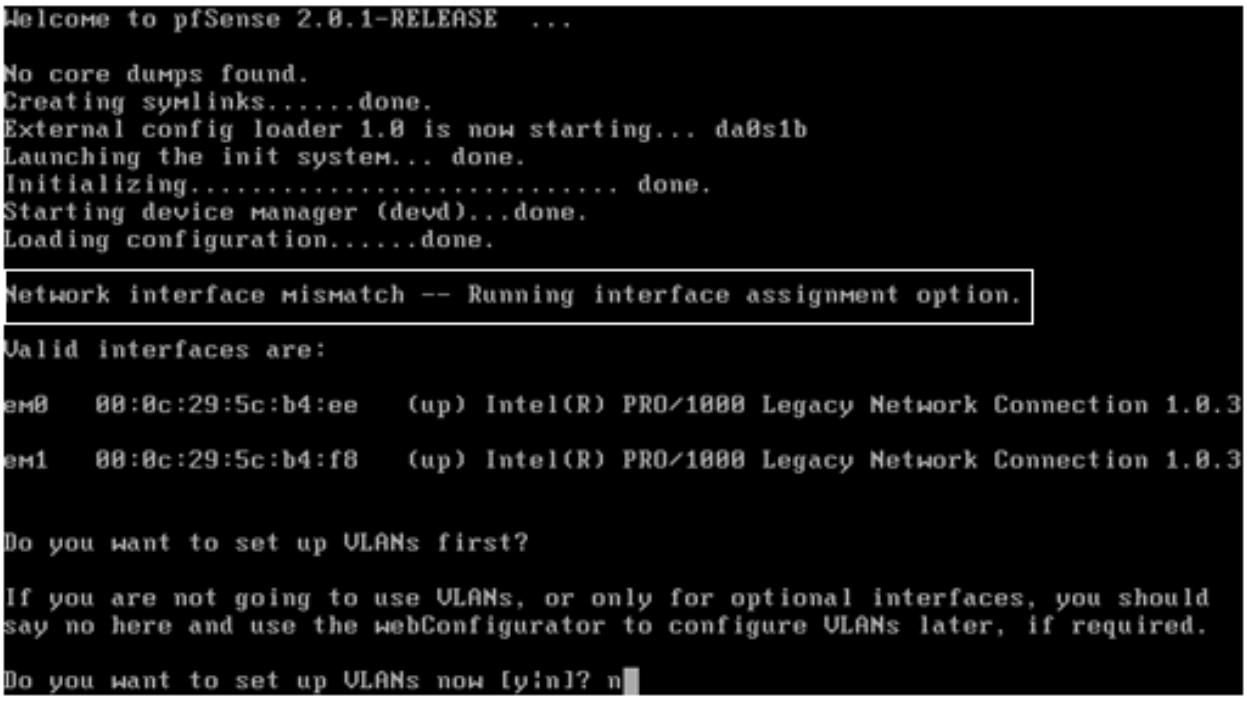
At this point the pfSense virtual machine will reboot and **the CD must be removed from the drive**.

Interface Assignment

Next up, the pfSense boot menu returns.



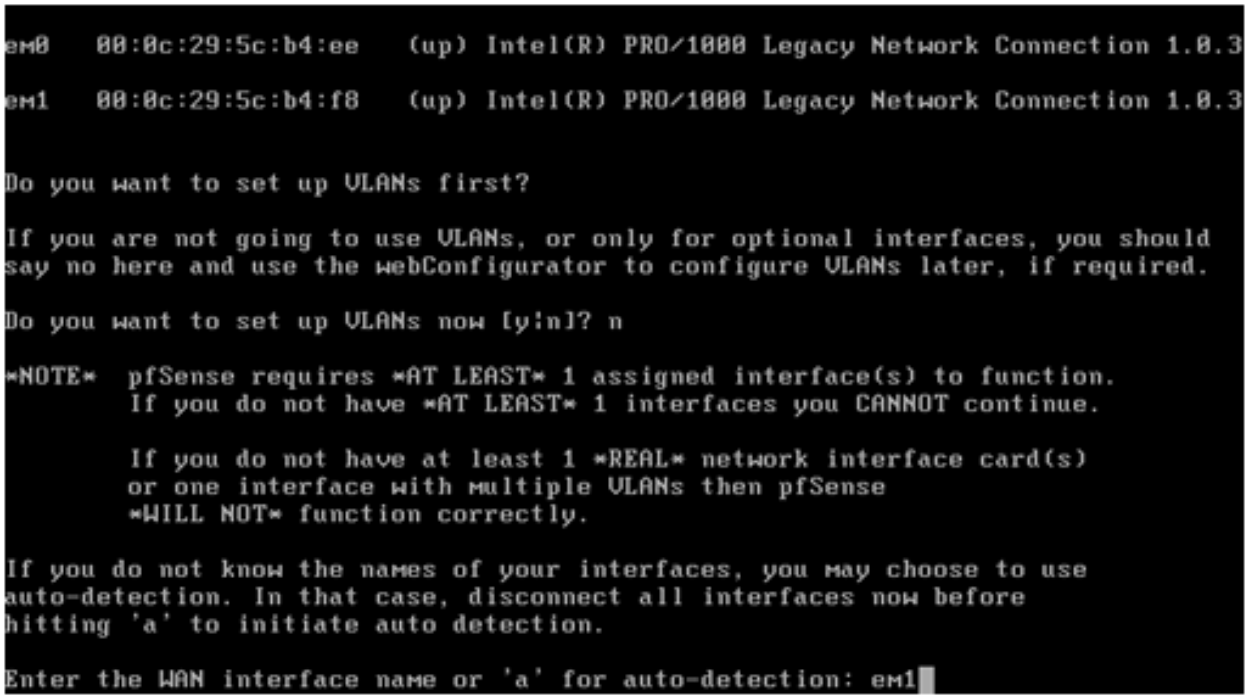
As the pfSense software is already installed on the virtual disk, allow the timer to expire.



Once the pfSense software has booted the message: **Network interface mismatch – Running interface assignment option** is shown. This means that the pfSense instance has not yet been told which virtual network interface is LAN and which is WAN.

Note: As mentioned previously, on pfSense software version 2.2 and later, the choices in the default configuration are *em0* for WAN and *em1* for LAN, so WAN should be assigned to NIC 1. This may differ from the screenshots shown here. The interface assignment prompt will no longer appear for hosts using *em* NICs, so be careful not to attach a LAN to *em0*!

First of all, though, as VLANs are not needed, type *n* and press **return**.



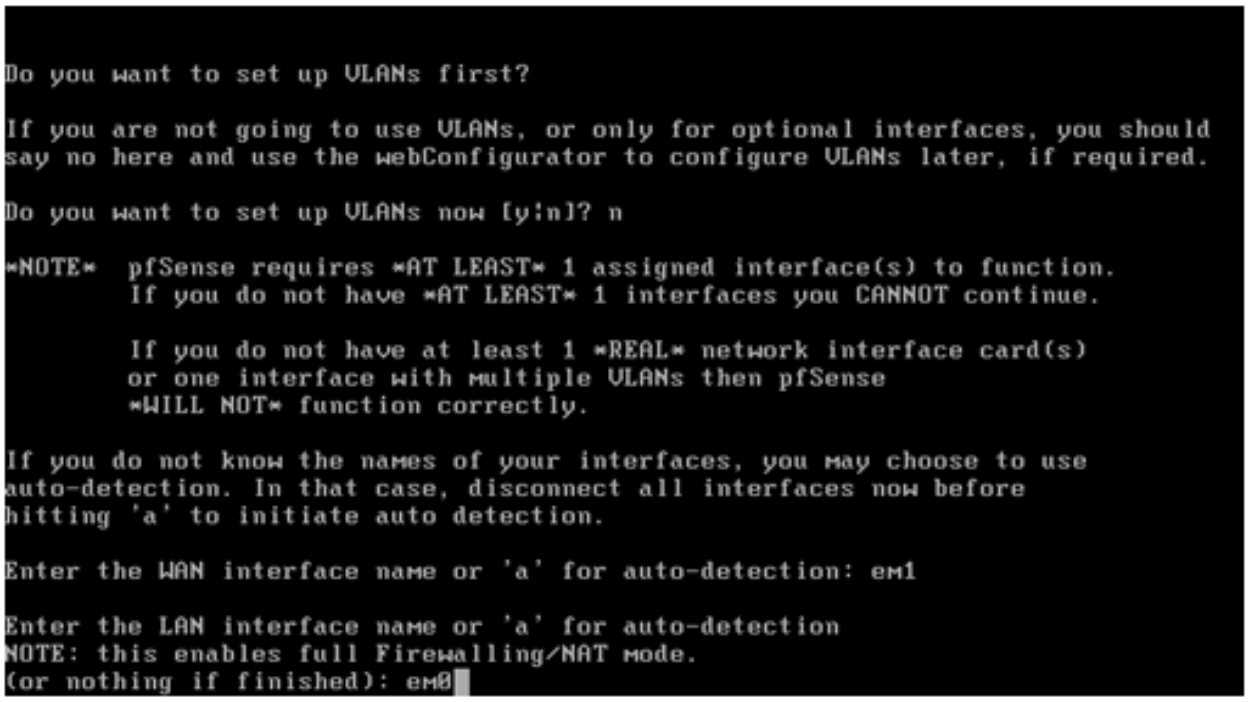
The order that the virtual NICs were assigned to the pfSense instance when the virtual machine was setup is important here. Generally, ESXi presents those network interfaces to the pfSense instance in sequence. That is, the pfSense virtual machine sees NIC 1 (WAN) as em0, NIC 2 (LAN) as em1, etc.

Note: The MAC addresses assigned to the virtual NICs and seen by the pfSense VM are also virtual. They are not the MAC addresses of the physical NICs.

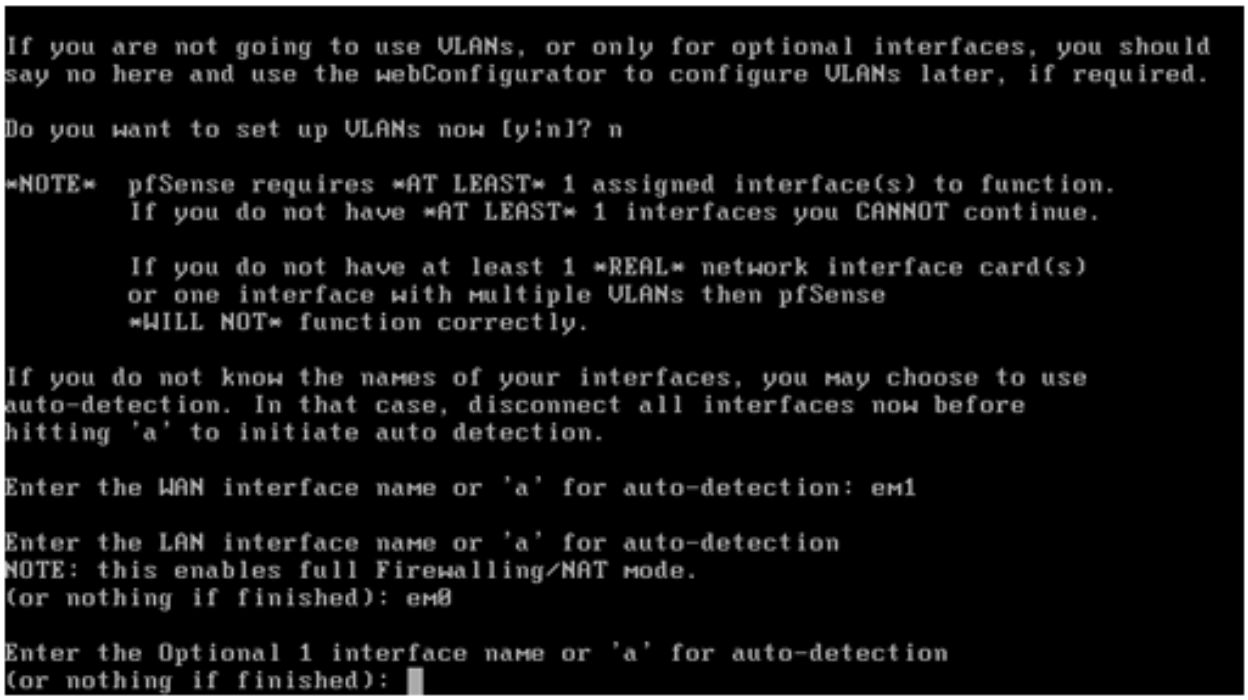
To double check which virtual NIC is which, right-click the virtual machine in the left-hand pane of the vSphere client and choose **Edit Settings**. Selecting each of the network adapters (WAN, LAN, etc) will show the virtual MAC address assigned to that interface. Make a note of these to help get the correct virtual interface assigned in the pfSense software. Only the last two characters of the vMAC are generally needed to match them against those shown in the pfSense console. For example:

WAN = ee LAN = f8

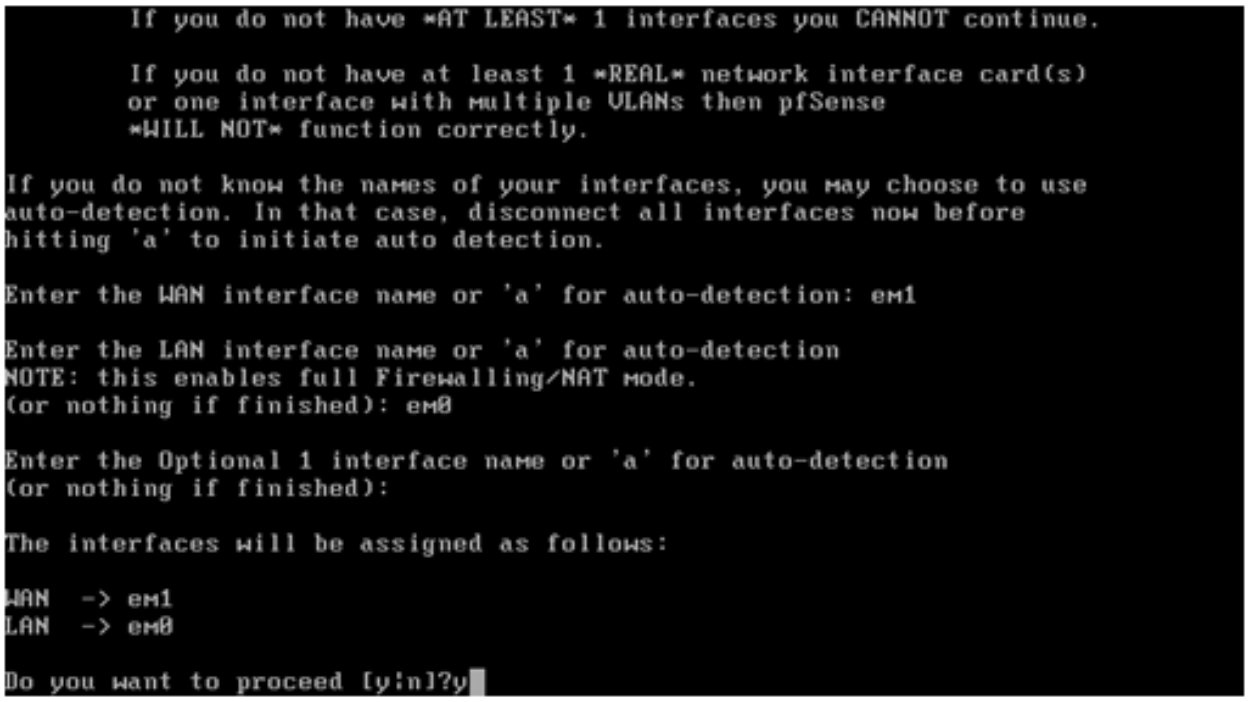
So go ahead and enter the WAN interface name, *em0* in this example, and press return.



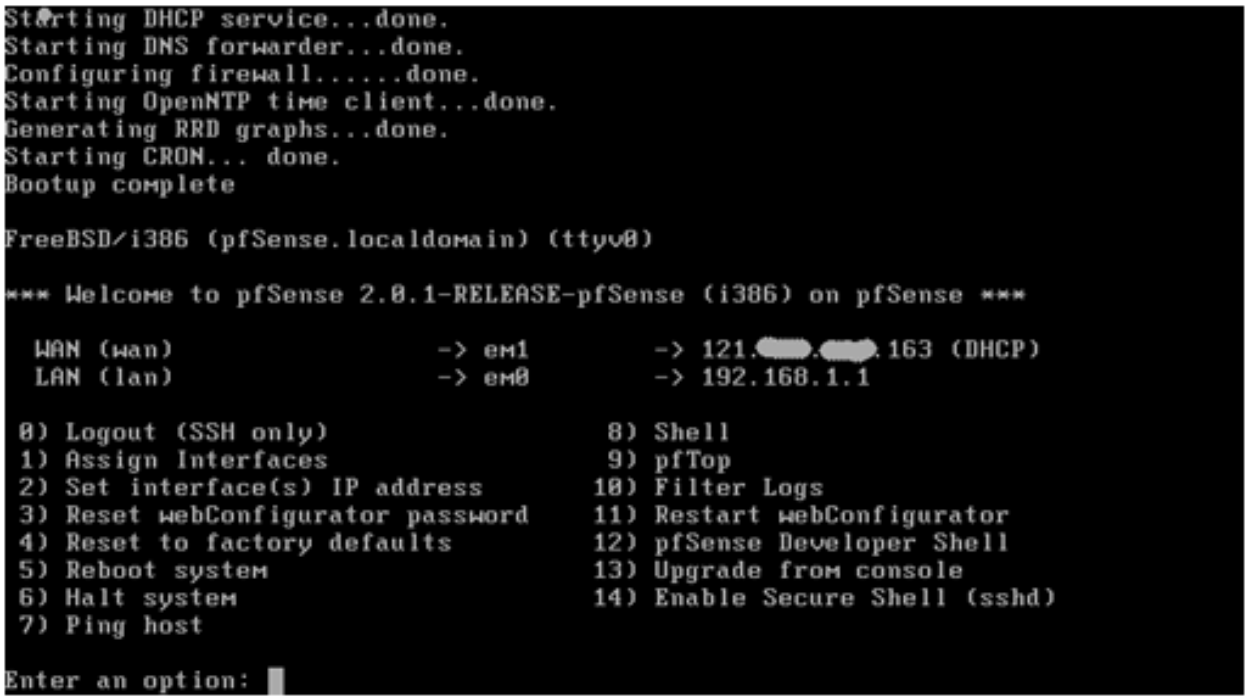
Now enter the LAN interface name, “em1” in this example, and press return.



As there are not any OPT interfaces, yet, press return.



Lastly, check that the interface assignments are correct, enter "y" and press return.



After a short interval, the pfSense VM will reconfigure itself, restart and present the main pfSense screen, above.

If the modem (in this example a simple cable modem) is connected to the physical WAN port of the ESXi host, the WAN interface should have received a public IP address from the ISP via DHCP. ADSL and other modems may need to be set up to pass the public IP through to the pfSense VM. Other types of WAN connections and configurations are beyond the scope of this article.

The LAN interface has its installation default IP address of 192.168.1.1. If another network address and/or subnet is

desired, it may be changed from the console or GUI.

34.71.4 Adding a DMZ

Having a WAN and a LAN is fine but perhaps another virtual machine will be added to the virtual network – maybe a mail server or a web server. After all, that is likely to be one of the reasons ESXi was used in the first place – as an alternative to running multiple physical machines.

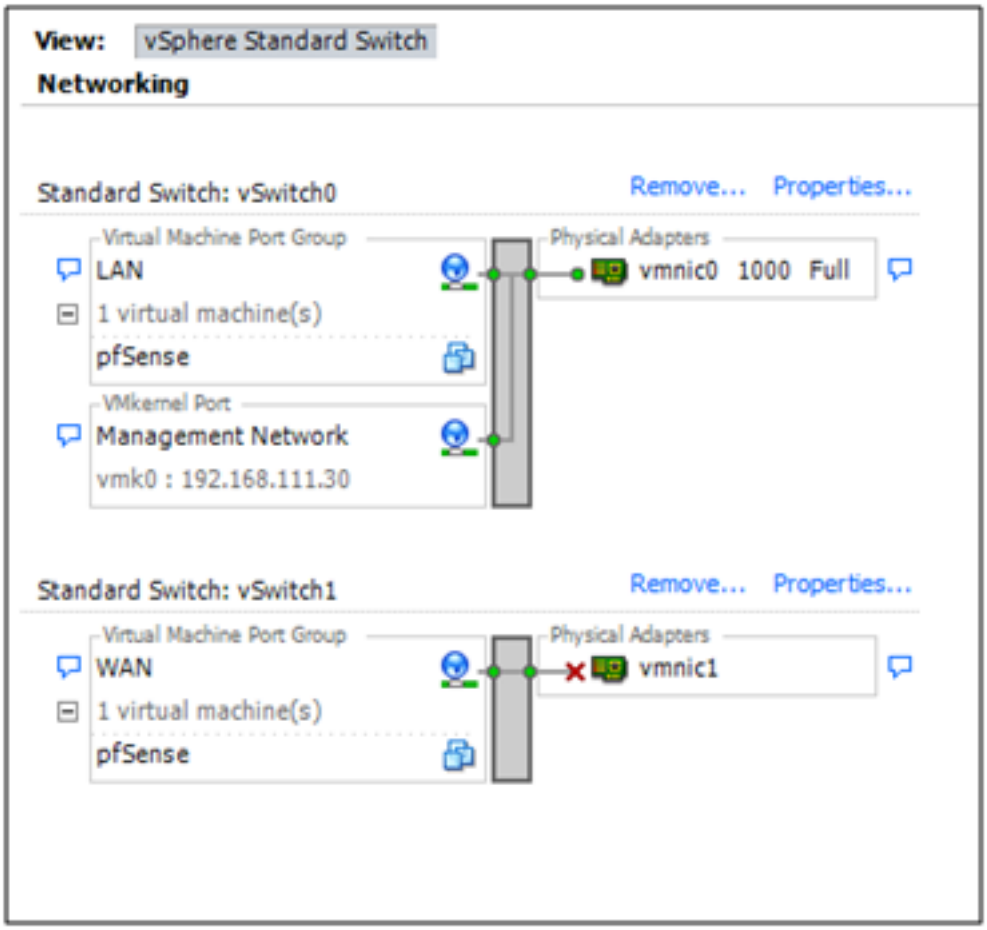
These kinds of servers should be accessible from the Internet but, at the same time, be protected behind the pfSense firewall. That way access can be controlled to them from both the LAN and the WAN.

Another interesting aspect of virtualization is that it is not necessary to stop at one DMZ. Because the DMZ network can be completely virtual, additional physical NICs are not required. For example, a virtual mail server could be put in one DMZ and a virtual web server in another. Then, by connecting them through the pfSense VM with virtual NICs, all access between the DMZs may be controlled. In addition, if one server is compromised, access to any of the others will be more difficult.

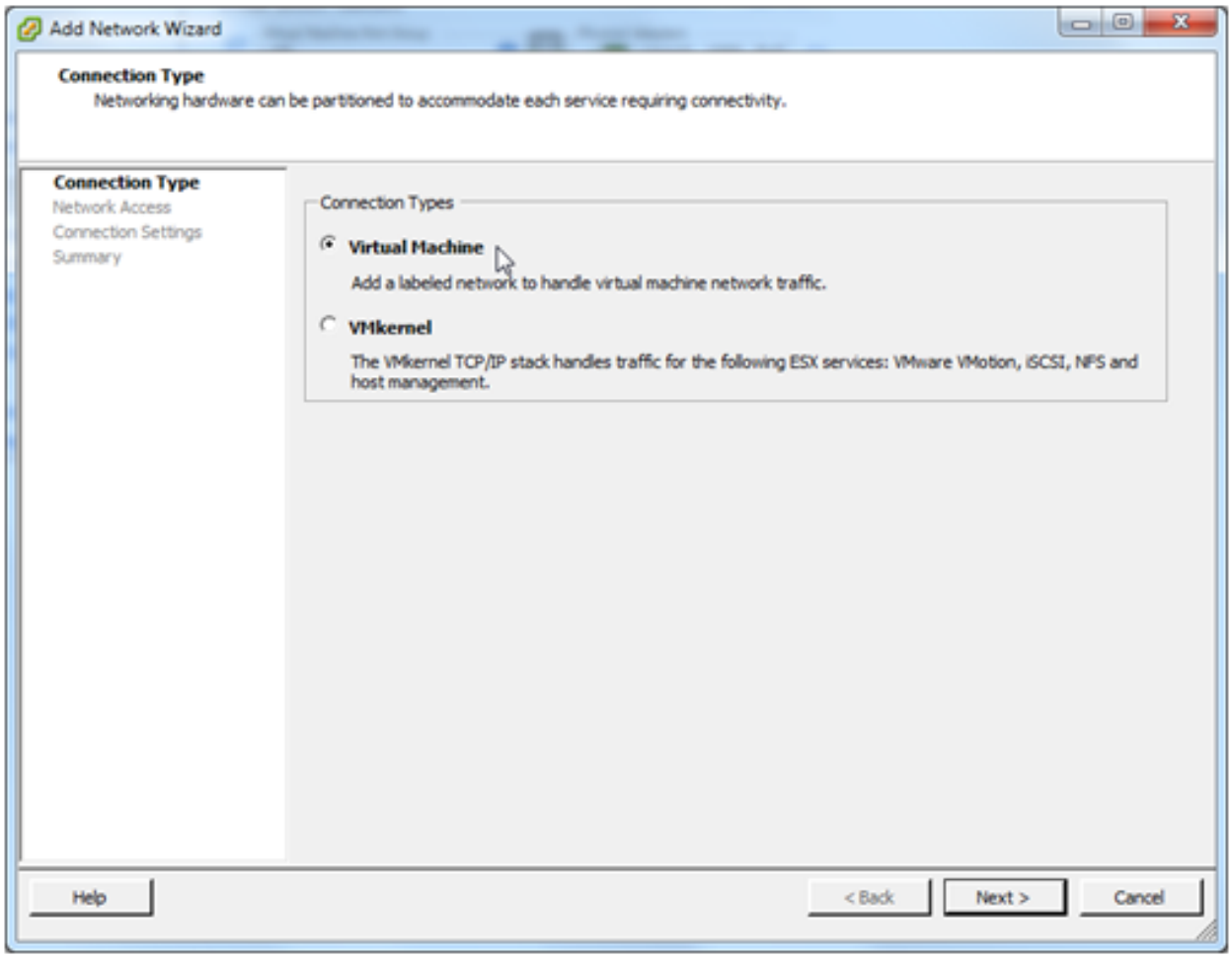
That's not to say that a DMZ can't also be connected to a real physical network as well. It may be desirable to connect a game console or video/music server behind the pfSense VM but not have it directly connected to the LAN. To accomplish that, connect a physical NIC to the ESX system and attach it as a DMZ.

Creating the DMZ network

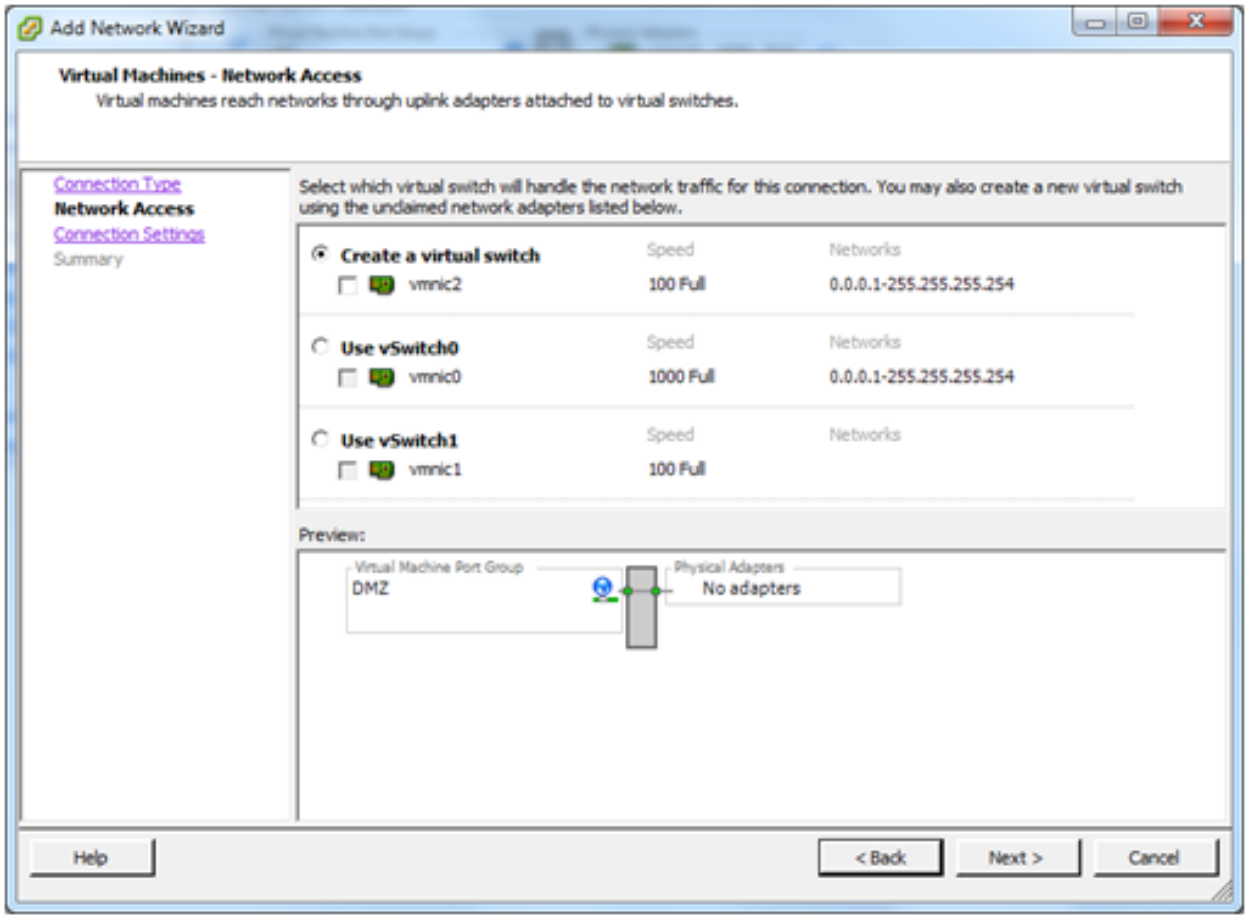
Go to the vSphere client and highlight the ESXi host. Click the Configuration tab and the Networking link. The ESXi network diagram is displayed.



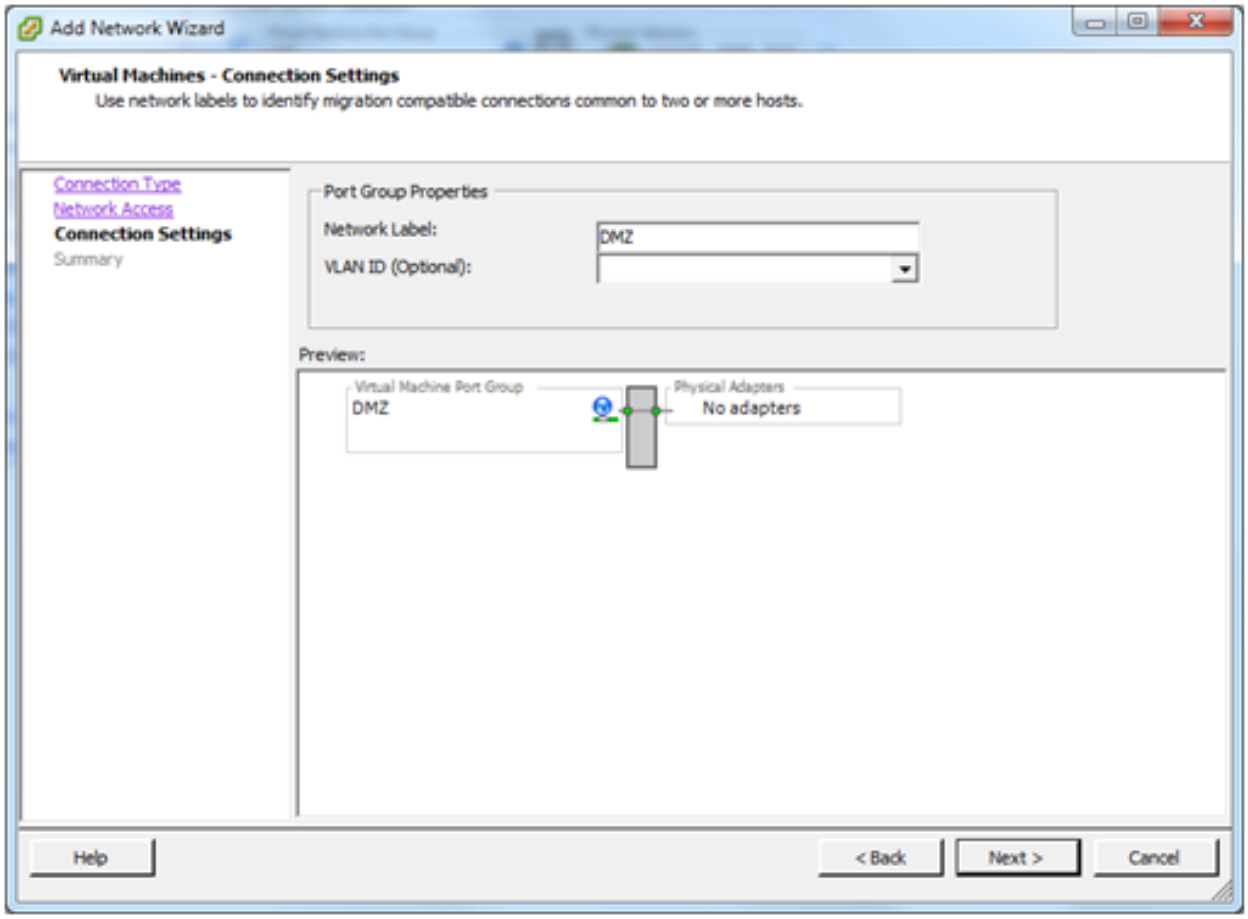
Click the Add Networking link near the top right of the Network pane.



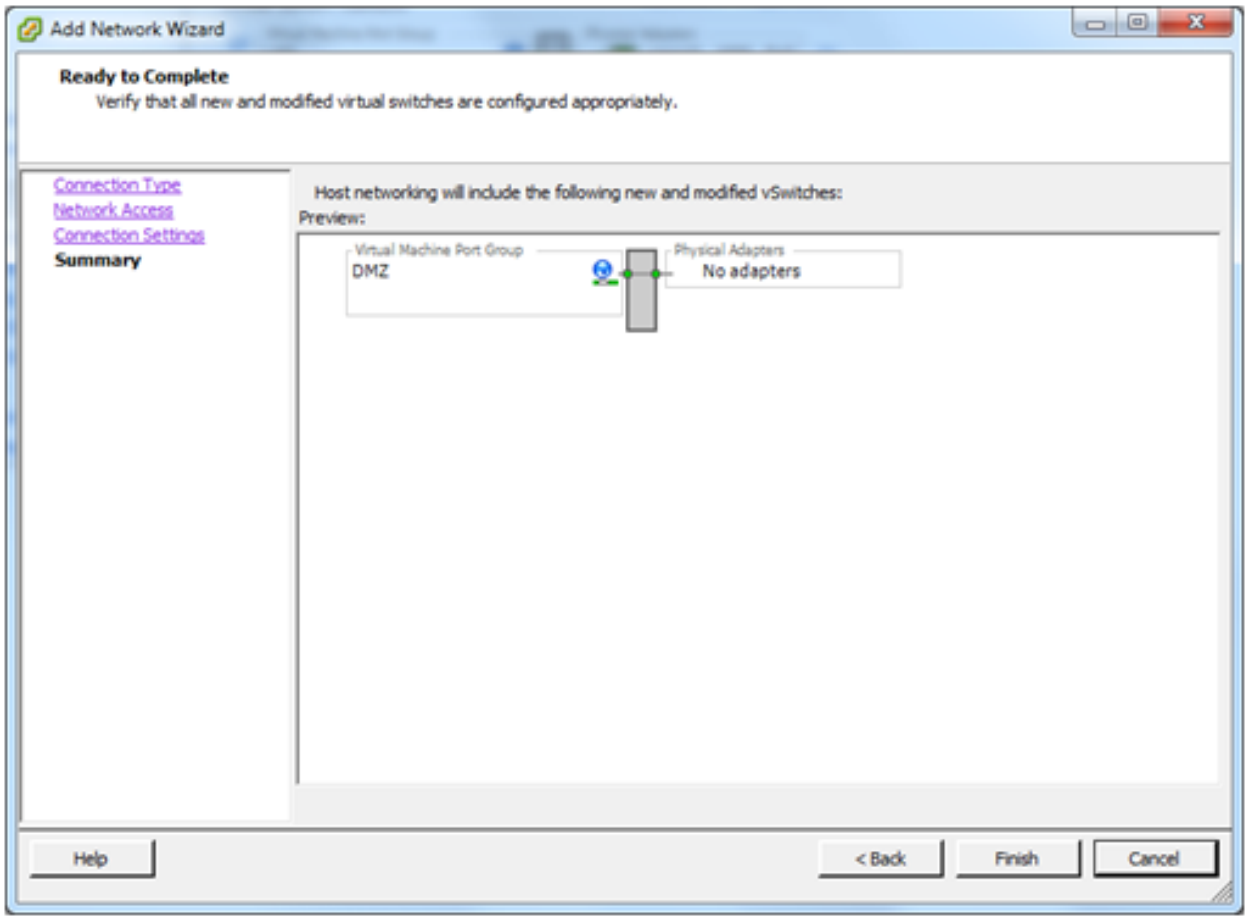
We want to add a new virtual machine network, so select that option and click Next.



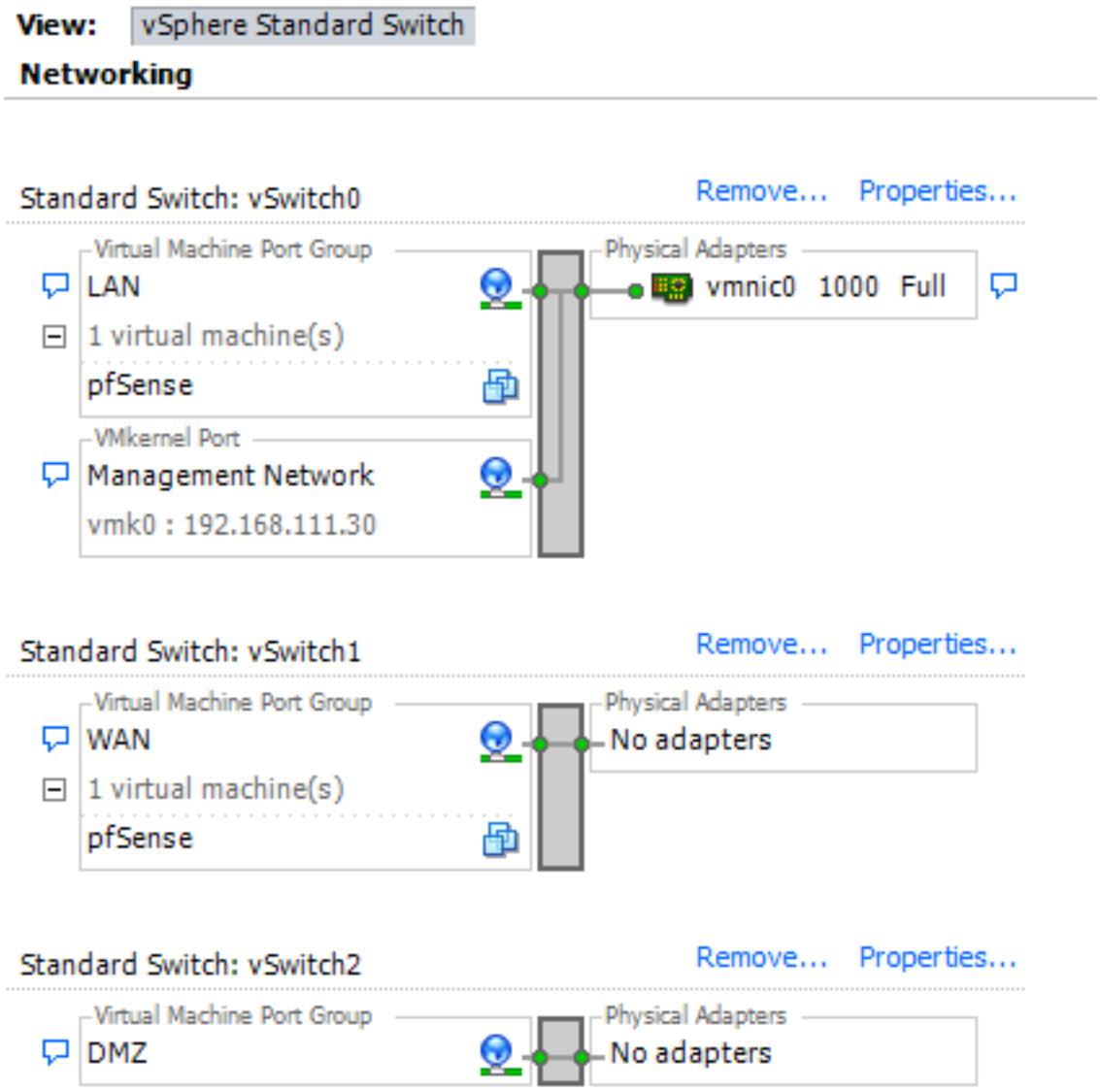
Choose the option to Create a vSphere standard switch. We aren't going to need a physical NIC – it is going to be virtual - so make sure that if there are more physical NICs in the ESXi host, none of them are selected, then click **Next**



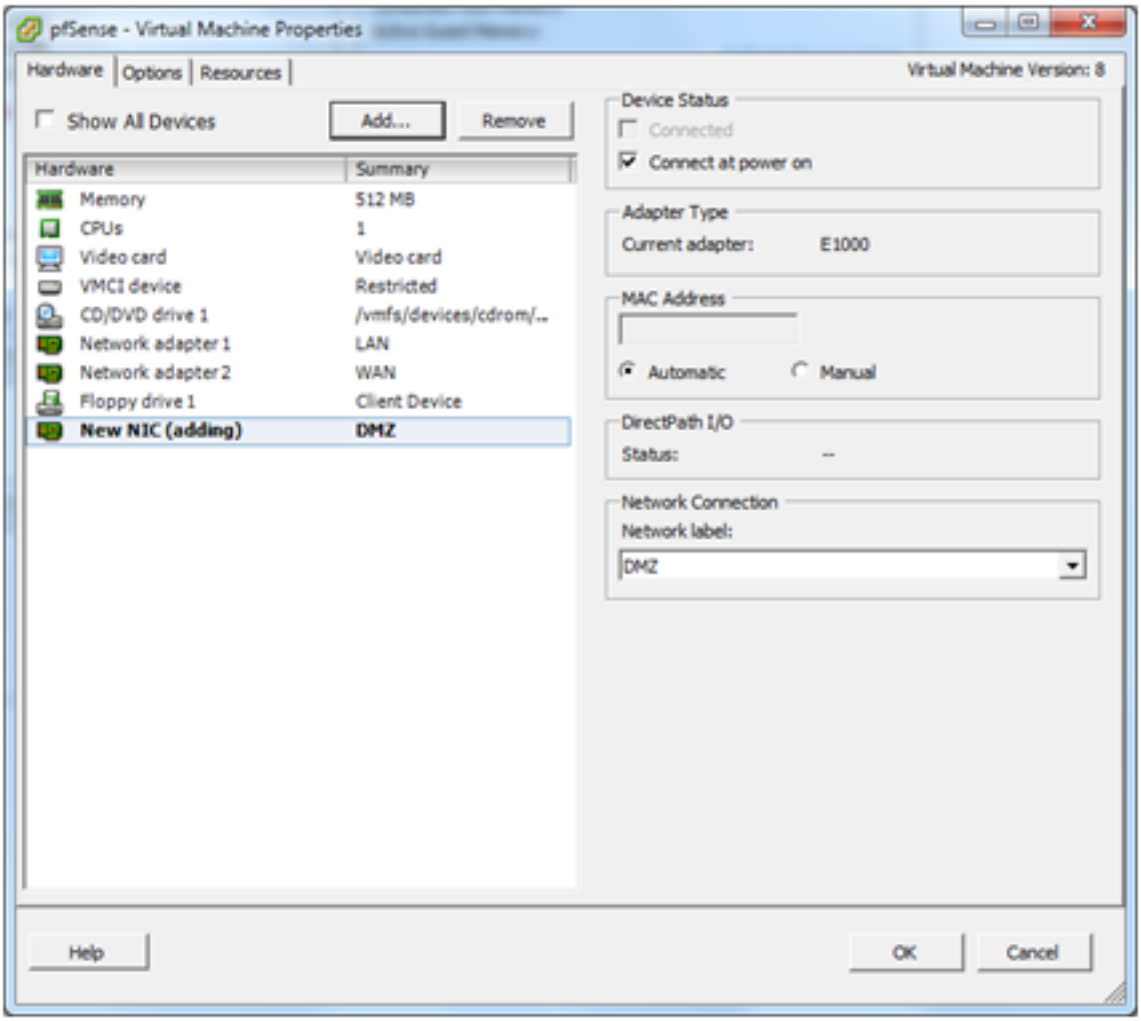
As with the LAN and WAN, give the new network a name. “DMZ” would be good. Click **Next**.



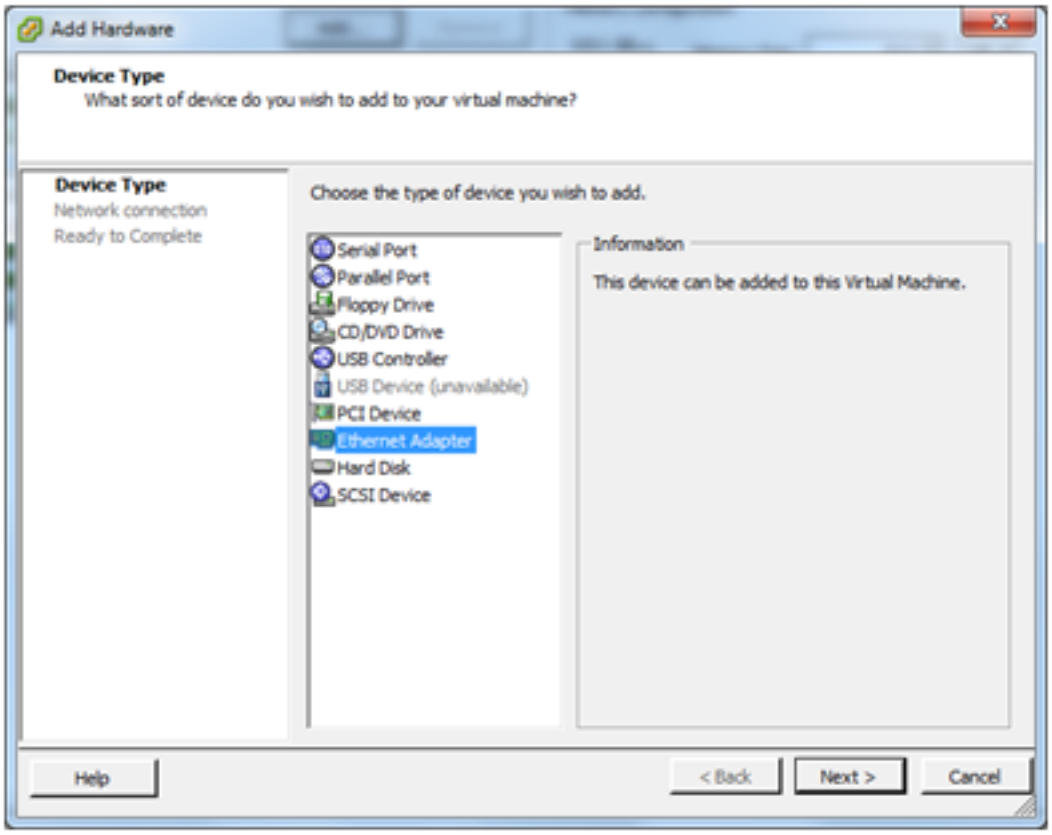
Click Finish.



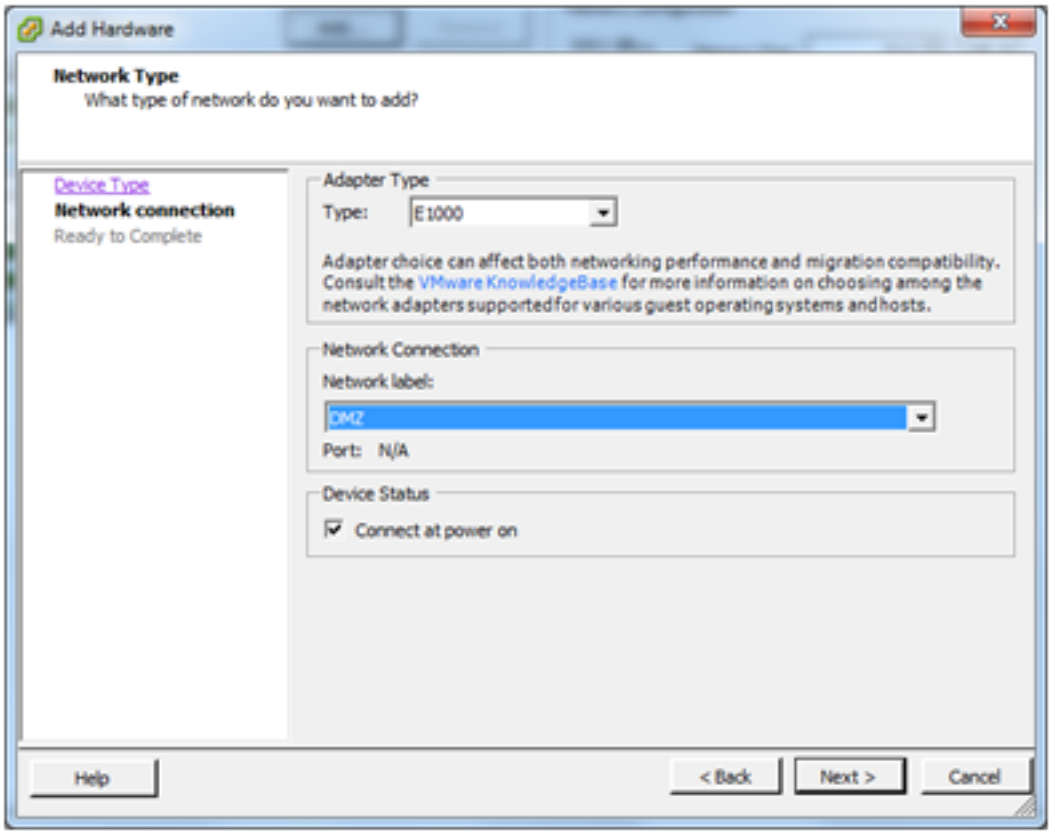
Now the Networking diagram will look like this - just a vSwitch and a Virtual Machine Port group called “DMZ” with no physical NICs attached.



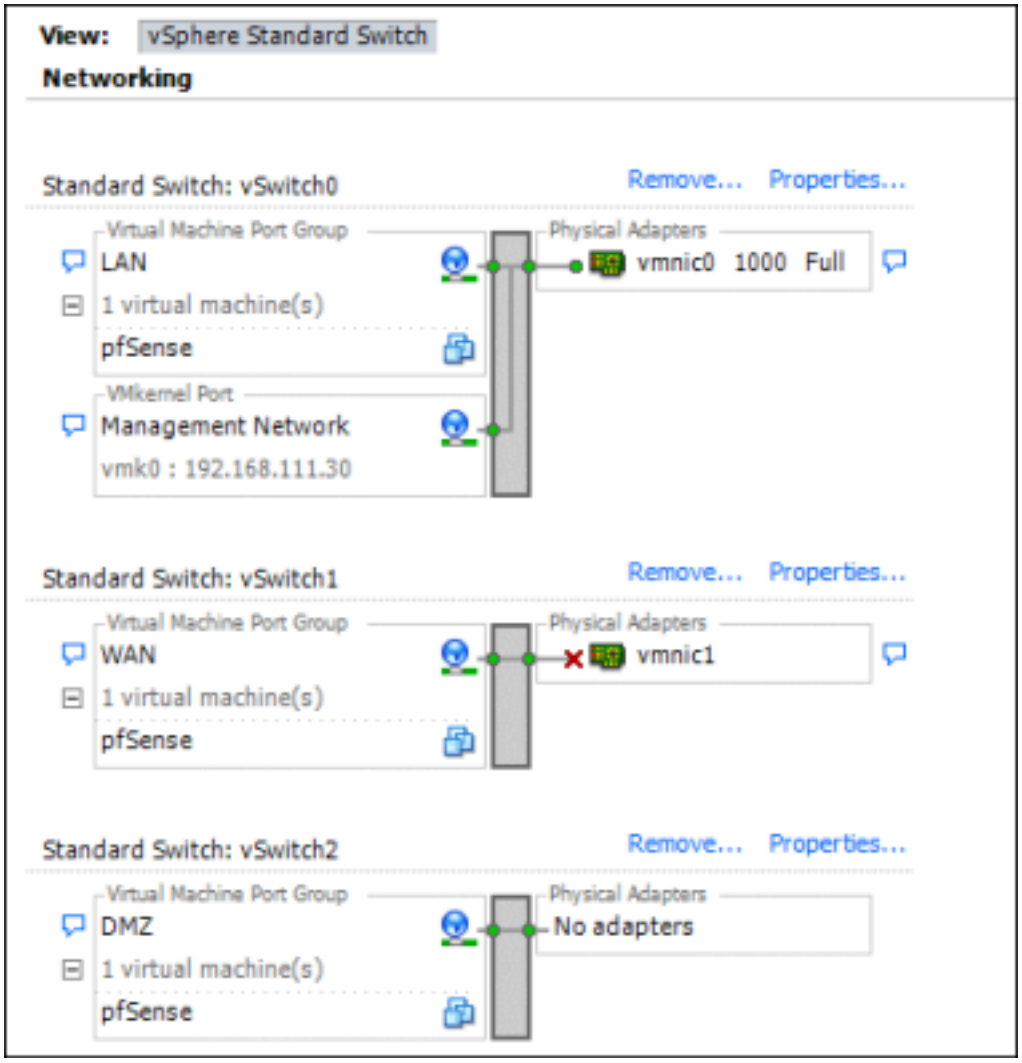
The next step is to connect the pfSense VM to this new DMZ network. Right-click the pfSense virtual machine and select **Edit Settings**. Click the **Add** button.



Choose **Ethernet adapter** and click Next.



As was done for WAN and LAN, choose the **E1000** type of virtual network adapter. Select **DMZ** from the drop-down list of available networks and choose **Connect at power on**. Click **Next**.



Now the network diagram should look like the above.

Note: all of this may be done while the pfSense virtual machine is still running. To make the pfSense VM aware of the changes, though, it will need to be restarted and then the interface must be assigned.

Now additional virtual machines may be attached to the DMZ network.

34.71.5 Installing VMware Tools

There are a number of benefits to installing the VMware tools, including better memory management, as well as improved network and disk performance. I can't vouch for those benefits but I find the most useful feature is the ability to shutdown or reboot a virtual machine without needing to log in to it directly. I use this to have all my VMs and the ESXi host gracefully shutdown in the event of a power outage that might exhaust the UPS battery . . . but that's another story.

The VMware Tools have been made available as a pfSense package, which makes the install very quick and easy.

Log in to the pfSense webGUI and click **System > Packages**.

From the **Available Packages** tab list, look for the **Open-VM-Tools** package and click  on the right to install the package. Confirm the the package installation and then it will proceed.

There is really nothing to configure with this package, it should just work.

34.72 Virtualizing pfSense with Hyper-V

This article is about building and running a pfSense® virtual machine under Microsoft Hyper-V. The guide applies to any Hyper-V version, desktop or server (this includes the standalone Hyper-V Server). The guide explains how to install any major pfSense software version under Hyper-V. Article covers the Hyper-V networking setup and pfSense software virtual machine setup process. The guide does not cover how to install Hyper-V or Windows Server. A basic, working, pfSense virtual machine will exist by the end of this article.

Note: If pfSense software will be used as a perimeter firewall for an organization and the "attack surface" should be minimized, many will say it is preferable to run it non-virtualized on stand-alone hardware. That is a decision for the user and/or organization to make, however. Now back to the topic.

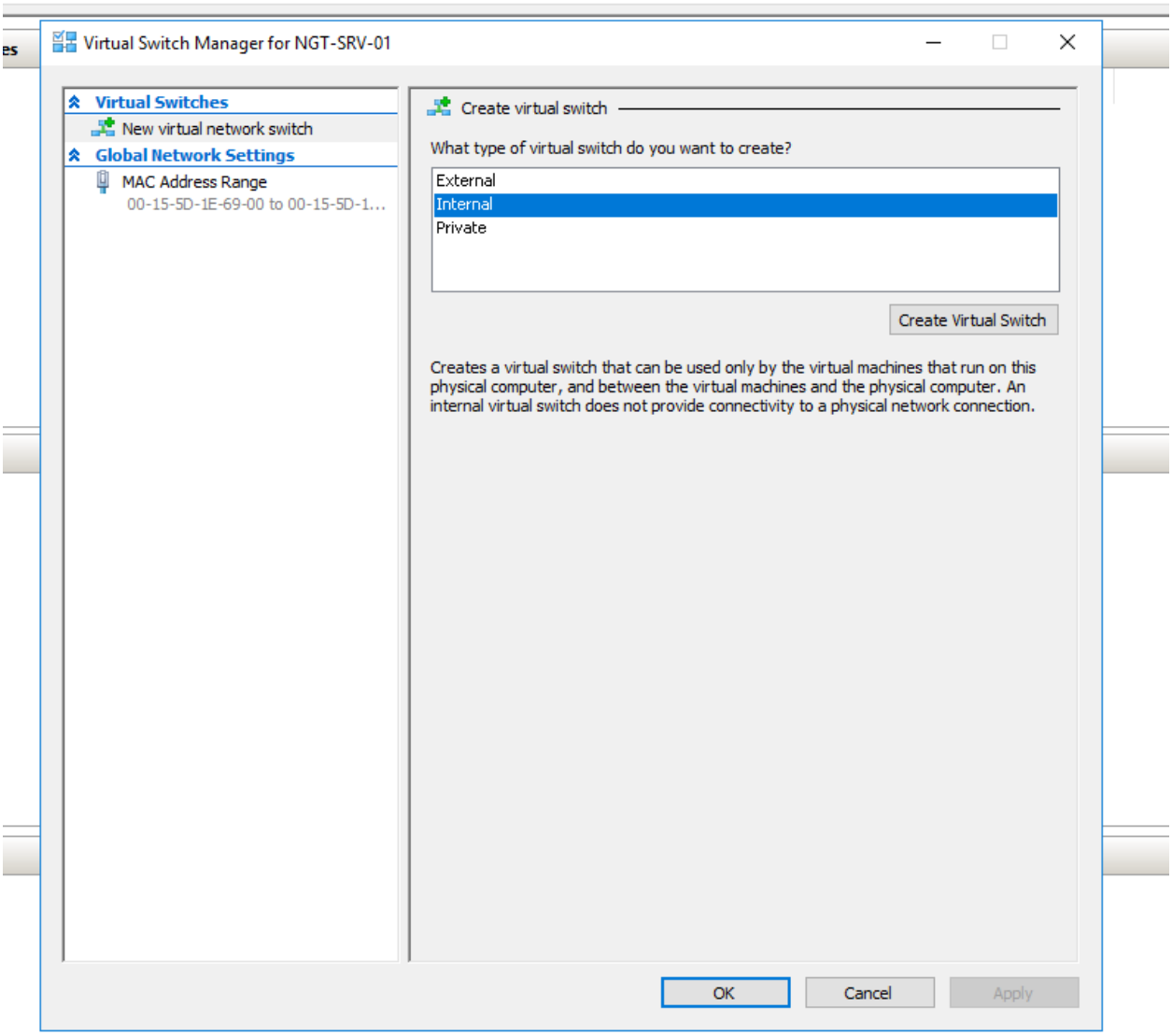
We're going to start at the point where we have a Windows Server 2016 with the Hyper-V role installed. If other VMs are already running on Hyper-V, then it is not likely necessary to follow the networking steps too closely. However, we recommend skimming through it to see what is suggested before building the pfSense virtual machine part.

34.72.1 Assumptions

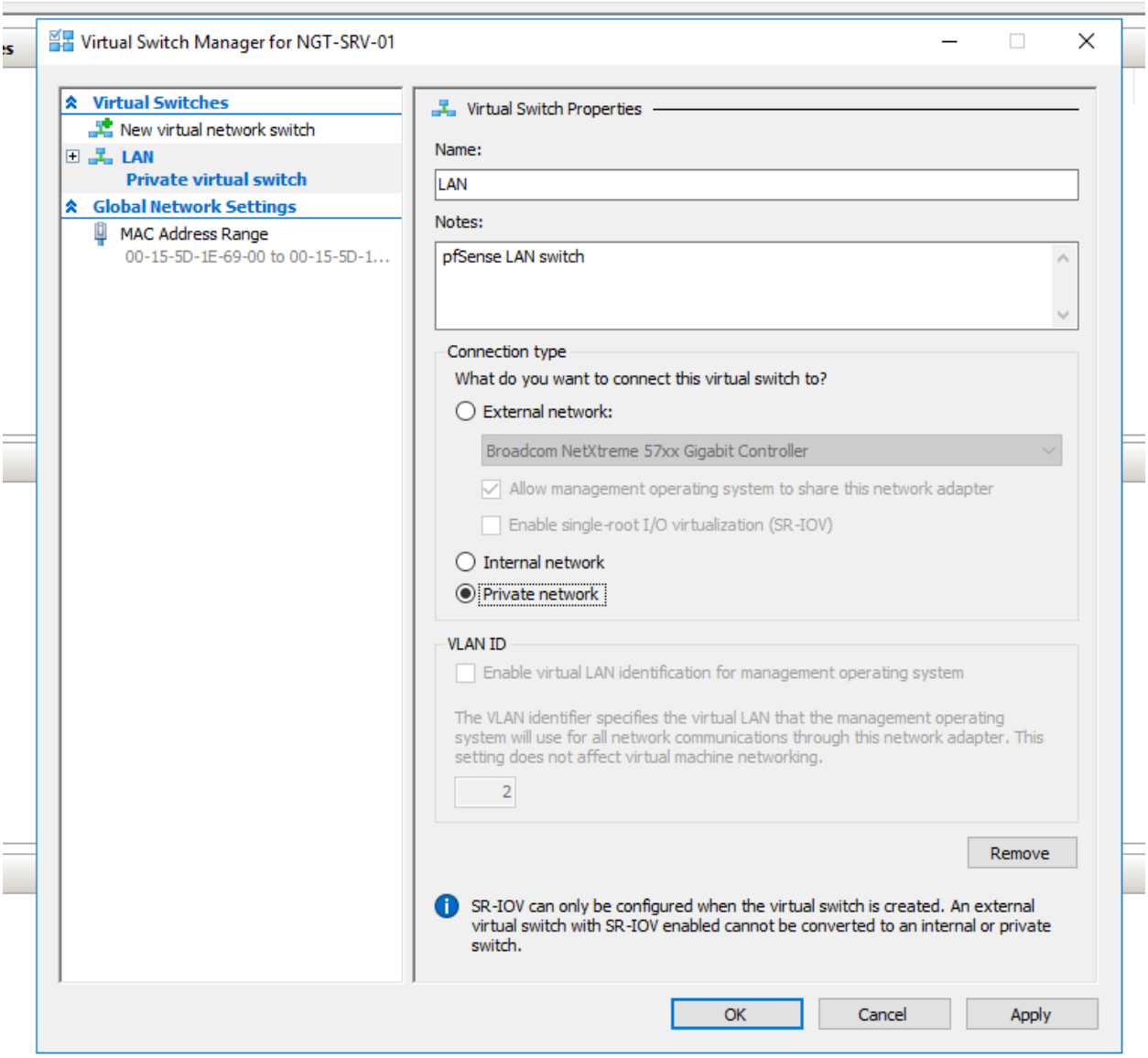
- Hyper-V host is up and Hyper-V role has been installed.
- The reader has an basic understanding of networking and Hyper-V virtualization

Basic Hyper-V Networking

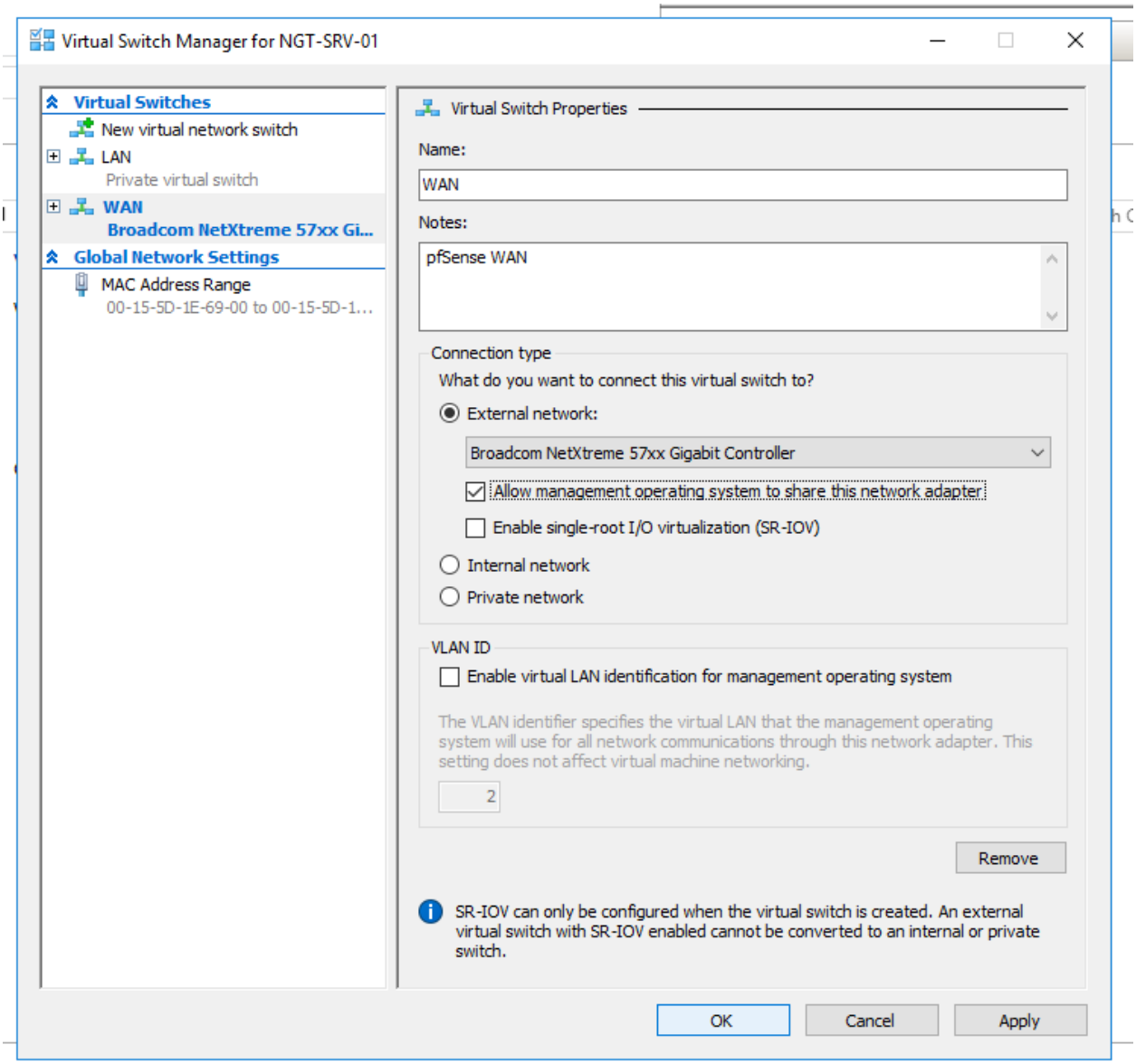
To virtualize pfSense software, first create two **Virtual Switches** via Hyper-V Manager. In the Hyper-V Manager open **Virtual Switch Manager** from the **Actions** menu. Select **Internal** type of virtual switch and click **Create Virtual Switch**



Name the newly added switch LAN and select private network. Click apply.

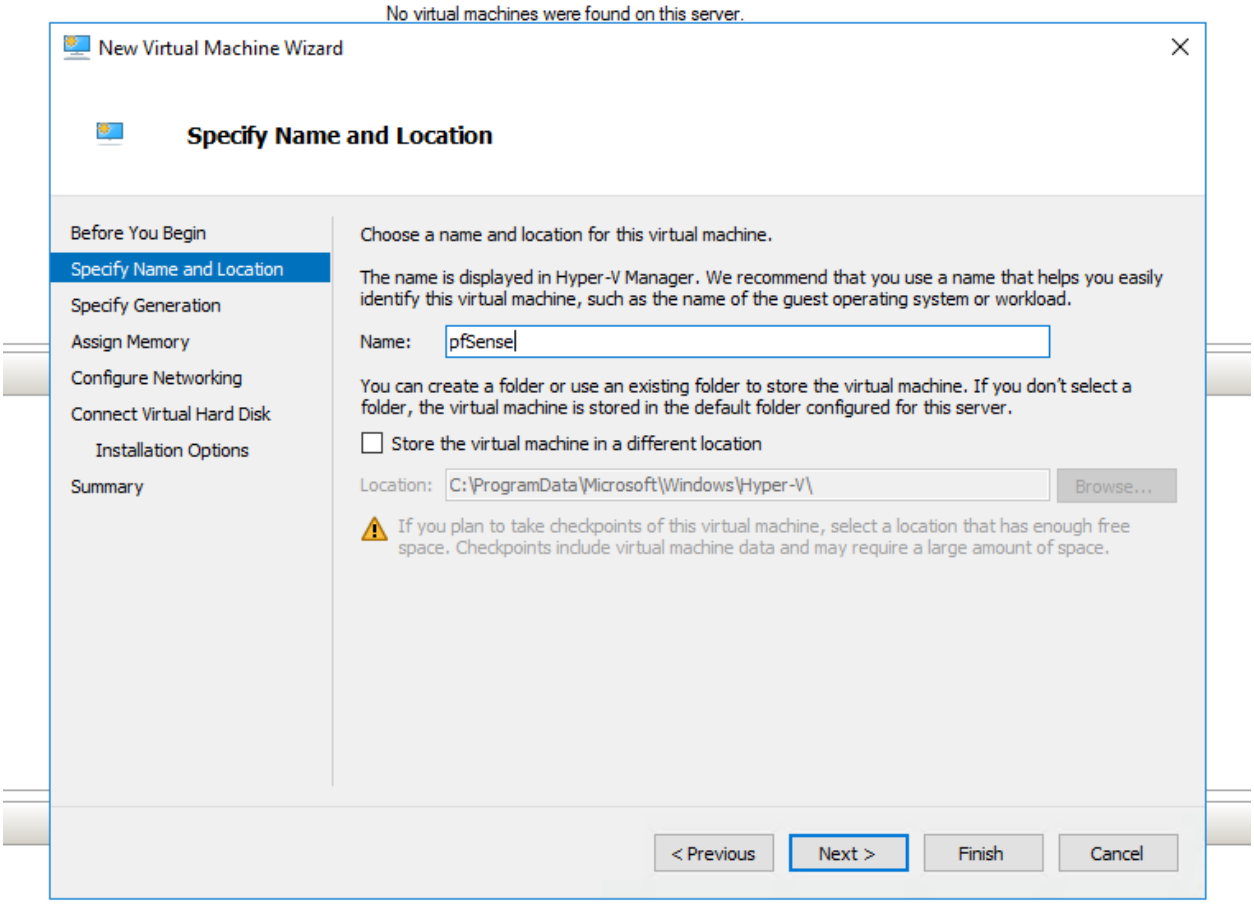


Now create WAN switch the same way as LAN. Make sure **Allow management operating system to share this network adapter** is not selected if the host has a dedicated NIC for WAN. For the purpose of this guide the management was allowed, however production use requires a separate NIC for WAN. Click OK.

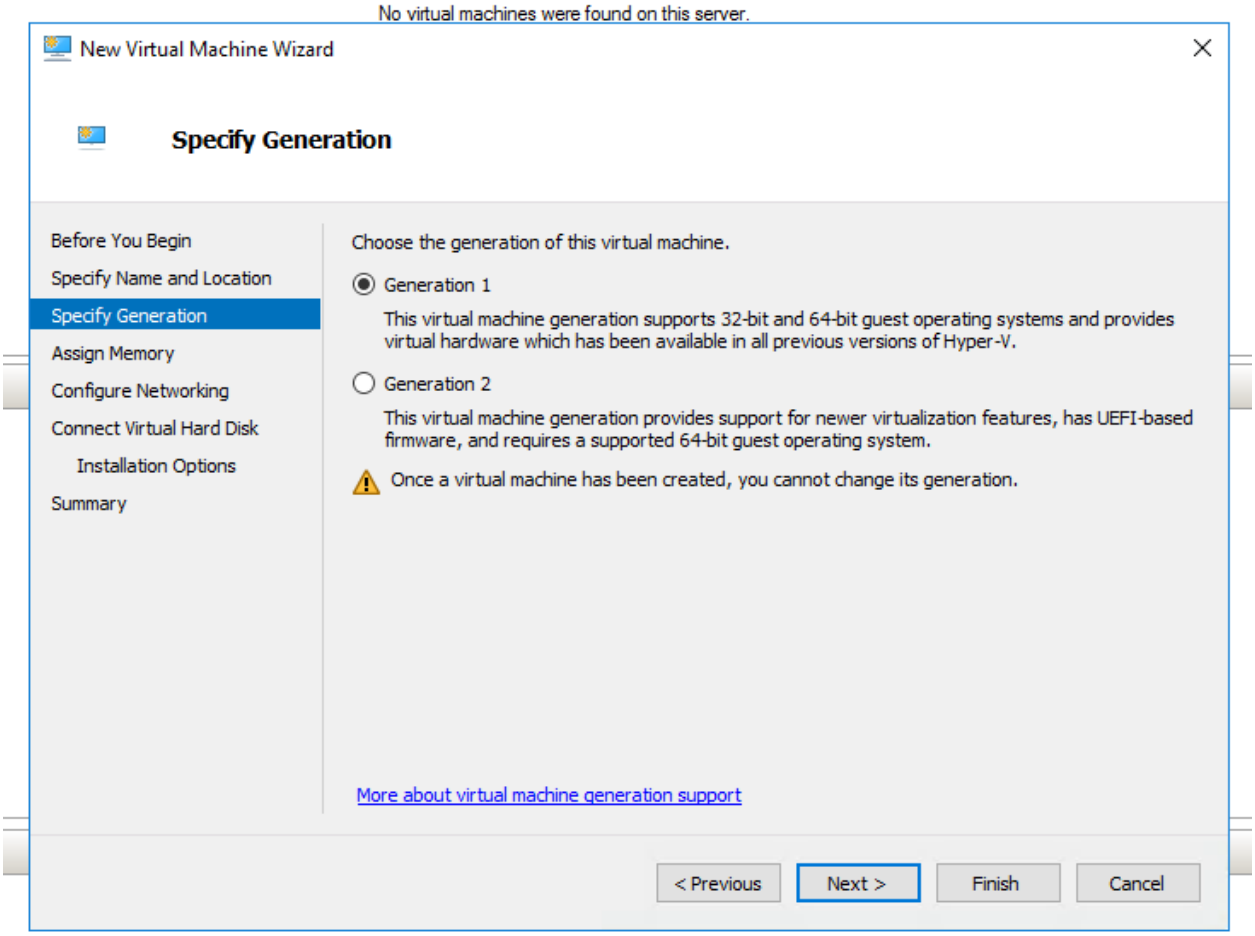


Creating the virtual machine

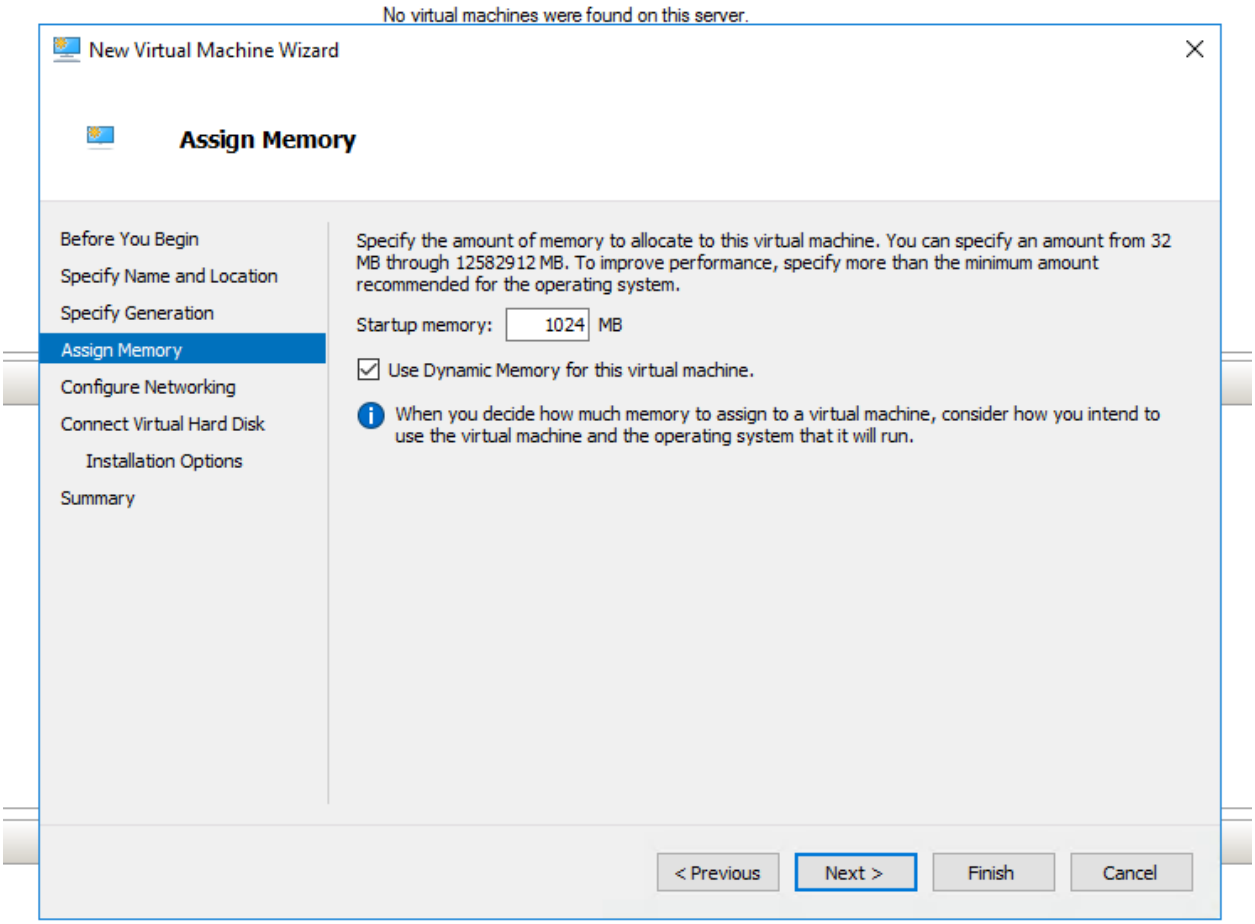
After creating WAN and LAN switches, we move to virtual machine creation. Start the new virtual machine wizard add a name.



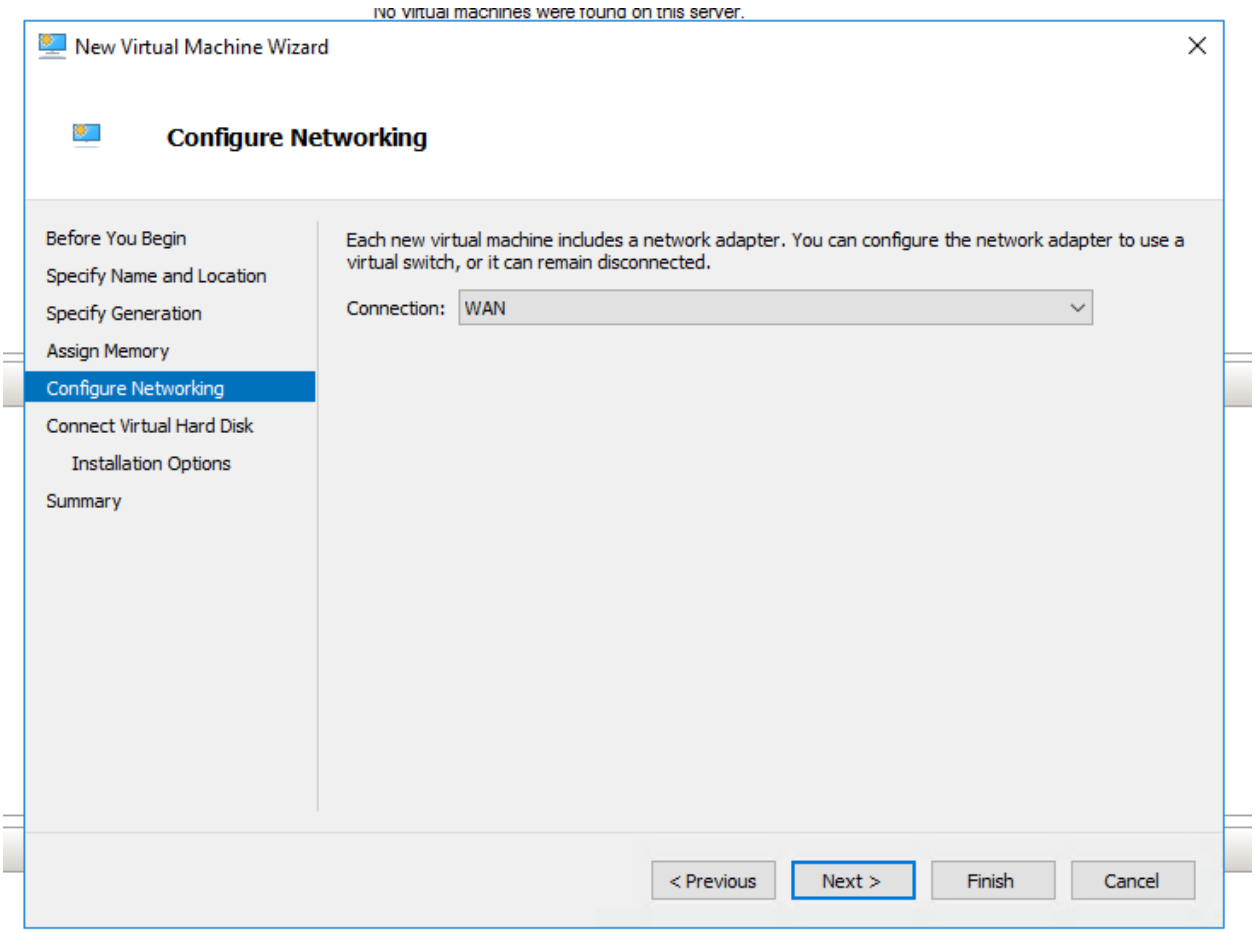
After clicking next select the appropriate virtual machine Generation: **Generation 2**.



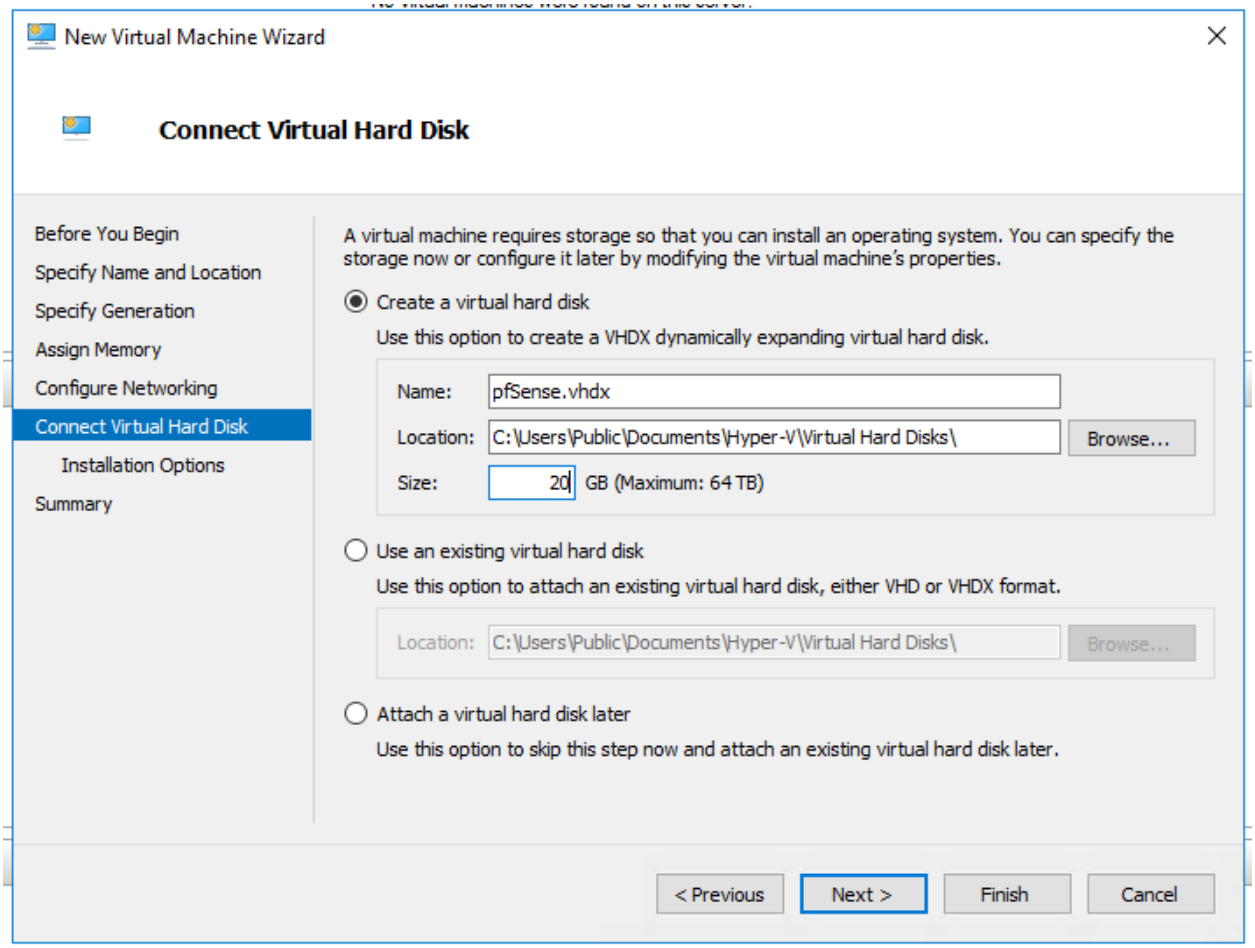
On the **Assign Memory** step add enough of RAM this deployment. This guide uses 1GB. 2GB is better if this VM will run multiple packages.



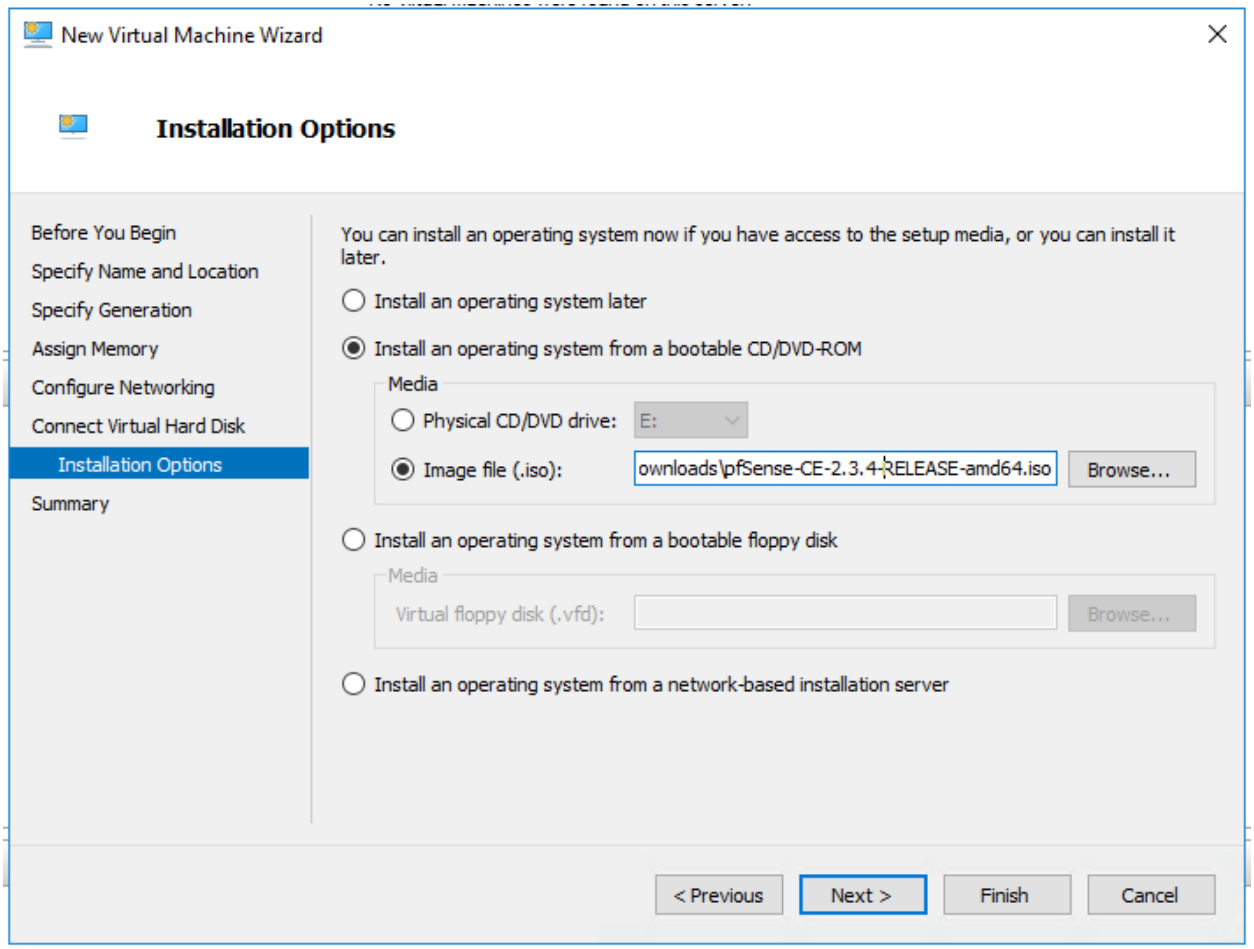
Next step is to **Configure Networking**, select WAN from **Connection** drop-down menu. We will add LAN later.



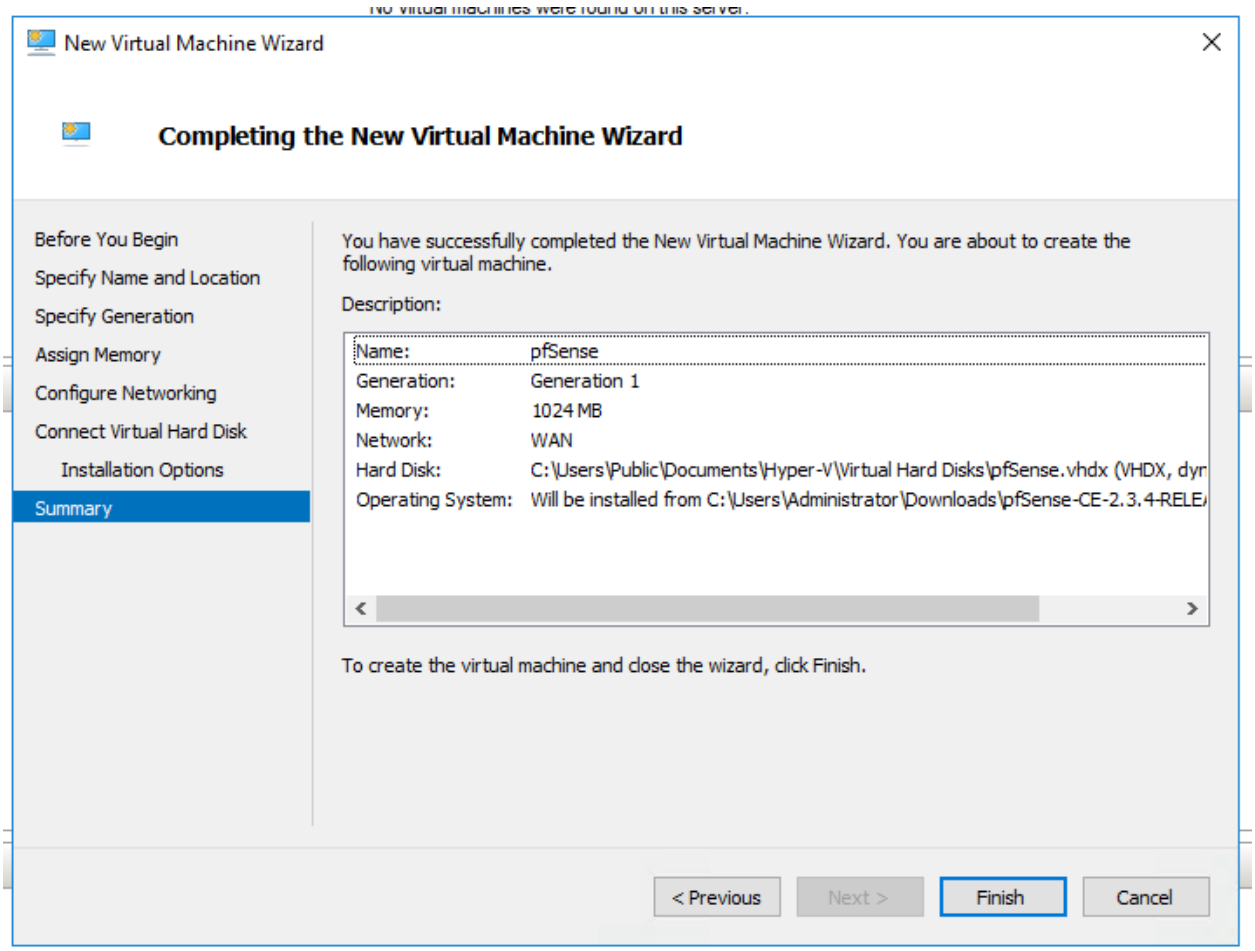
On the next step select **Create a virtual hard disk** and assign 10-20GB to the firewall. Larger disk size is required when running Squid caching.



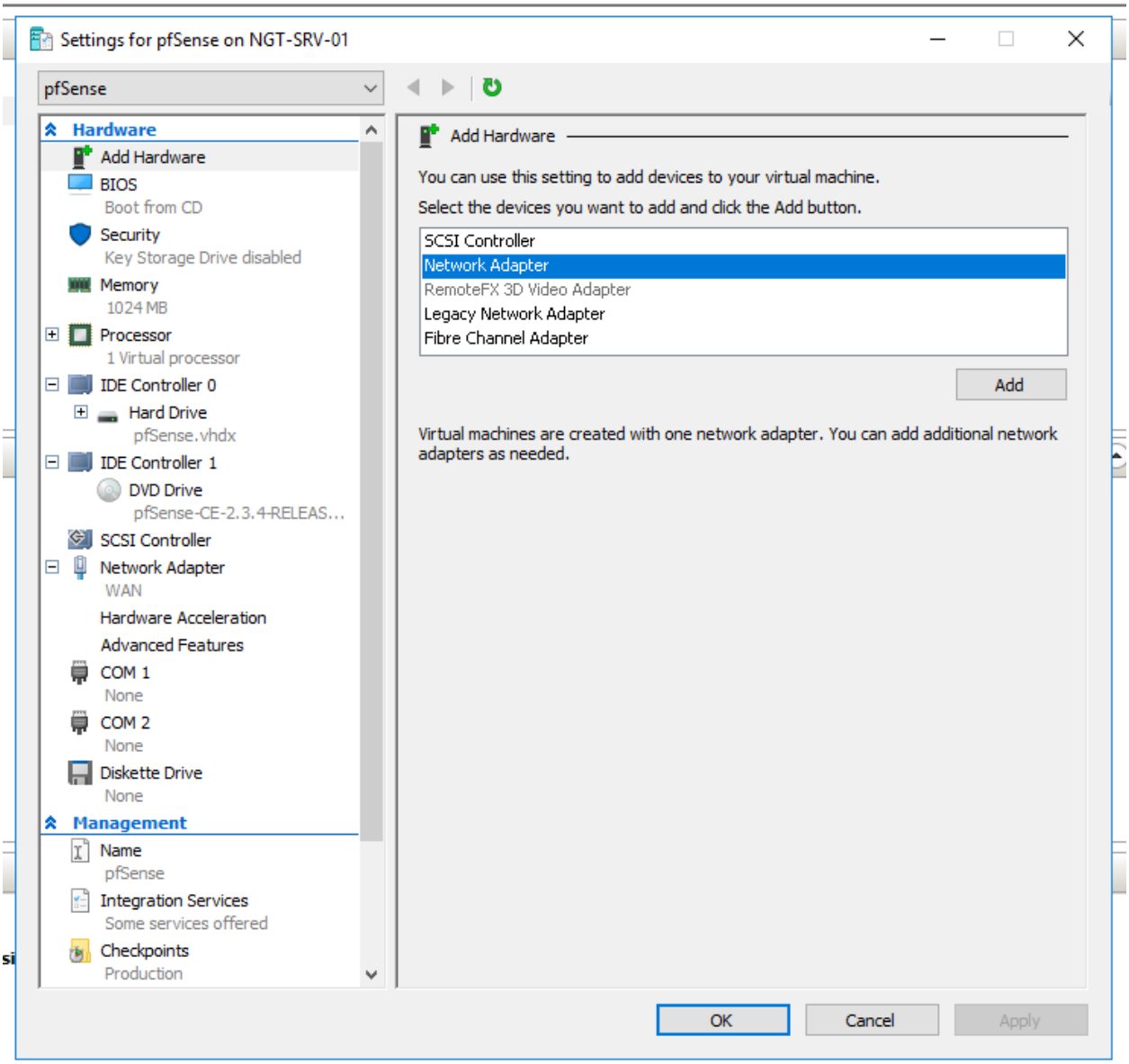
Select **Install an operating system from a bootable CD/DVD-ROM** and browse to the pfSense installer ISO.



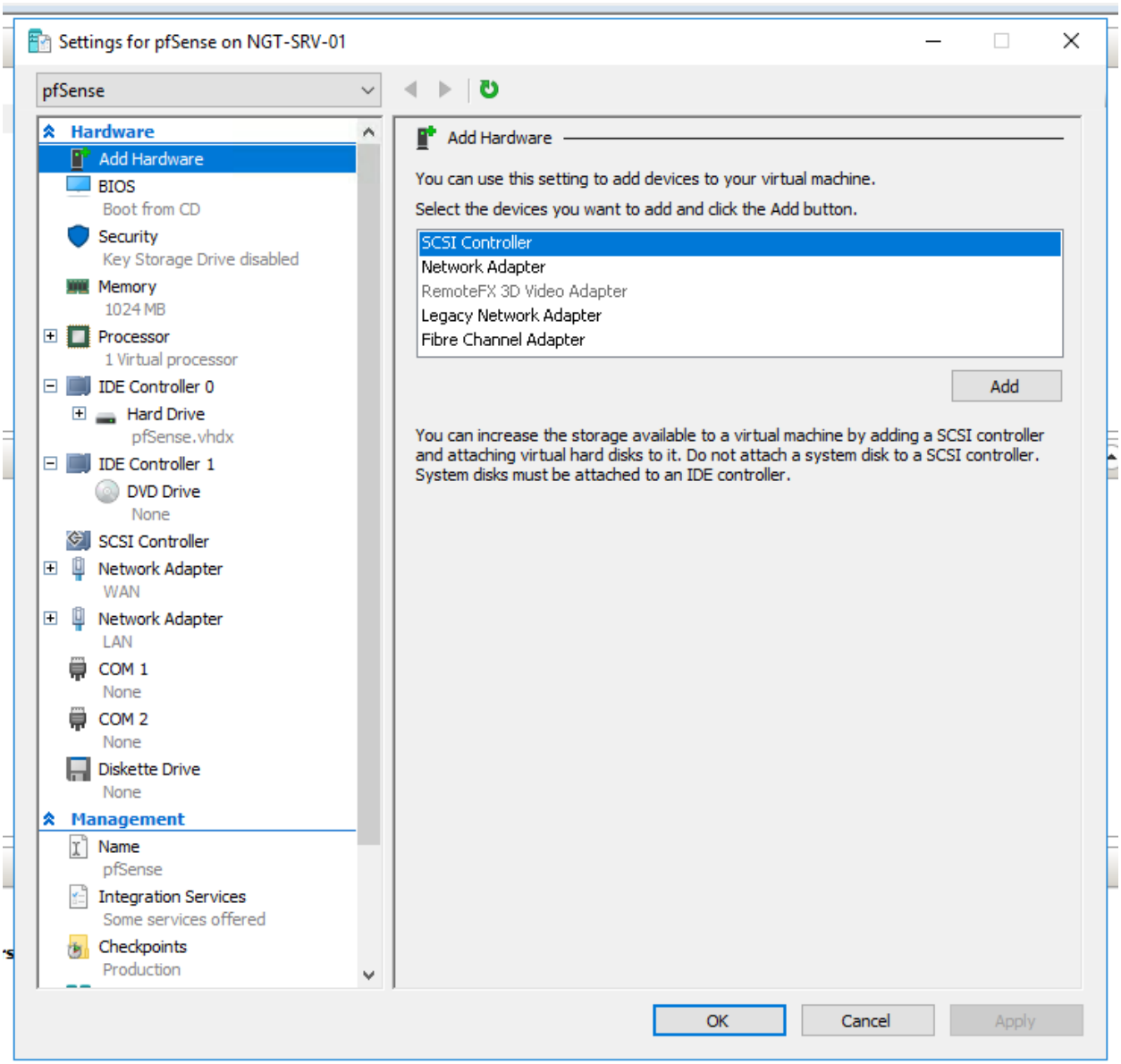
Review the virtual machine information and finish the wizard!



Open **Settings** of the newly created pfSense virtual machine and add another network adapter. Select **LAN** virtual switch for the adapter.

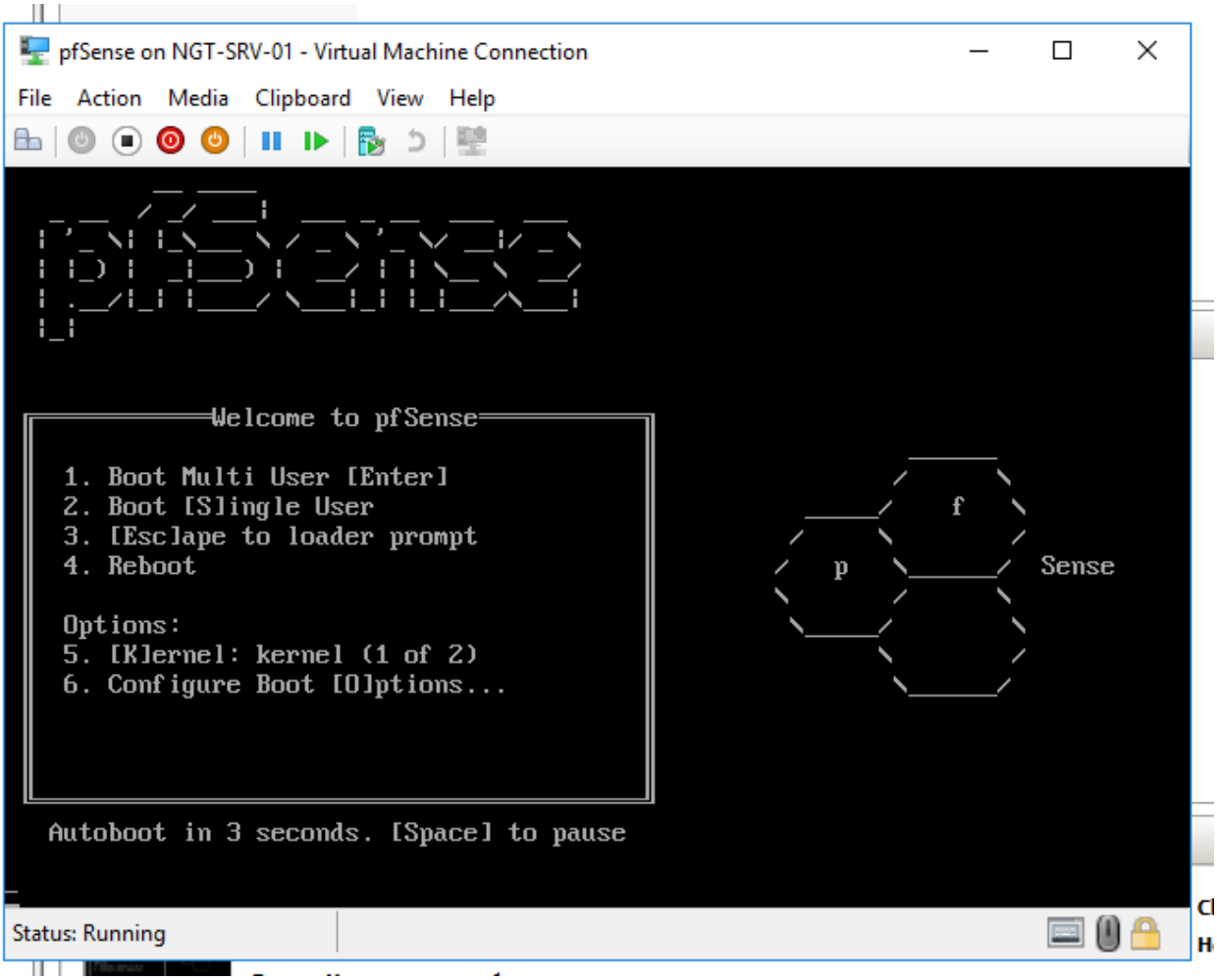


Review the VM settings and make sure to have WAN and LAN switches selected under network adapters

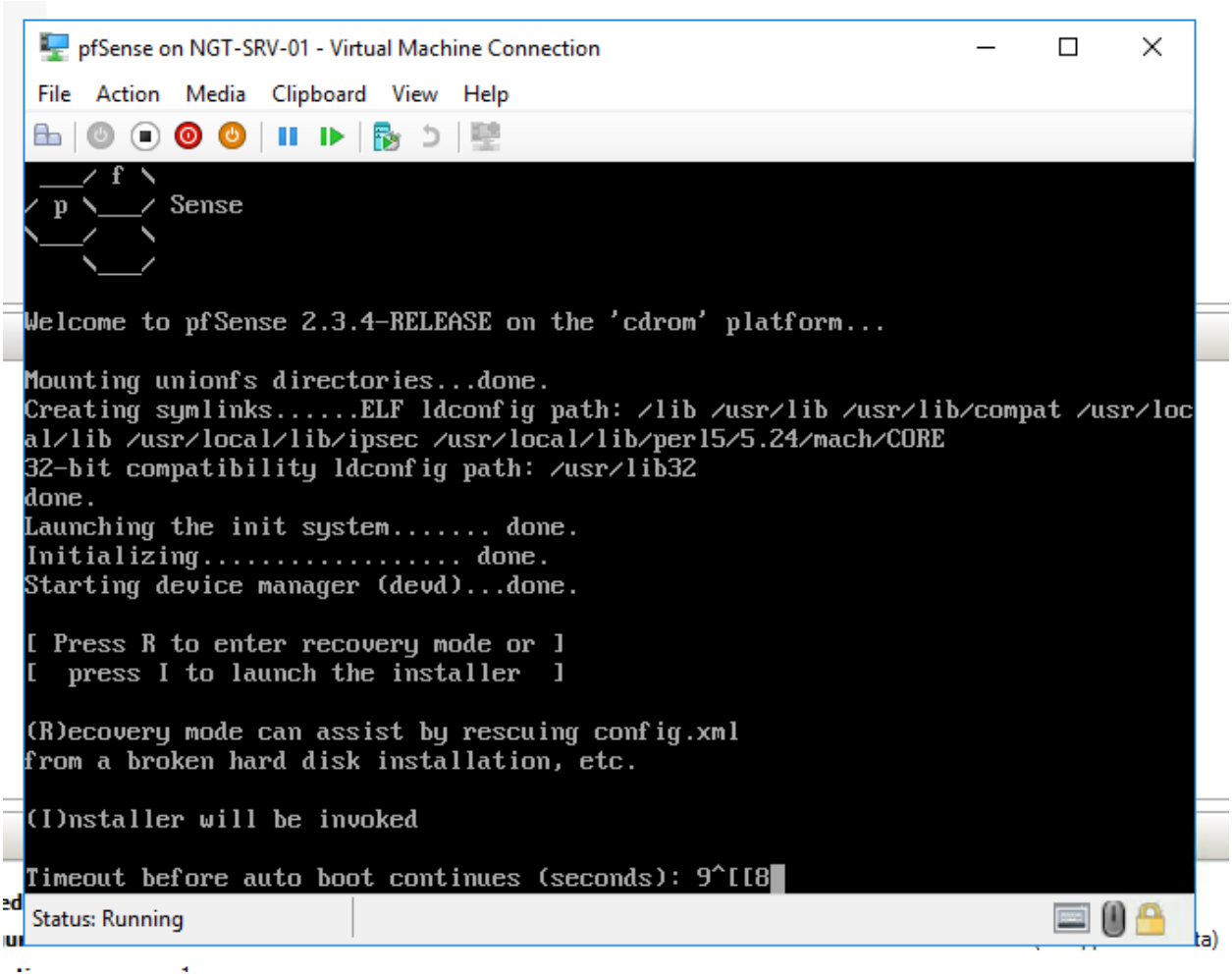


Installing pfSense Software

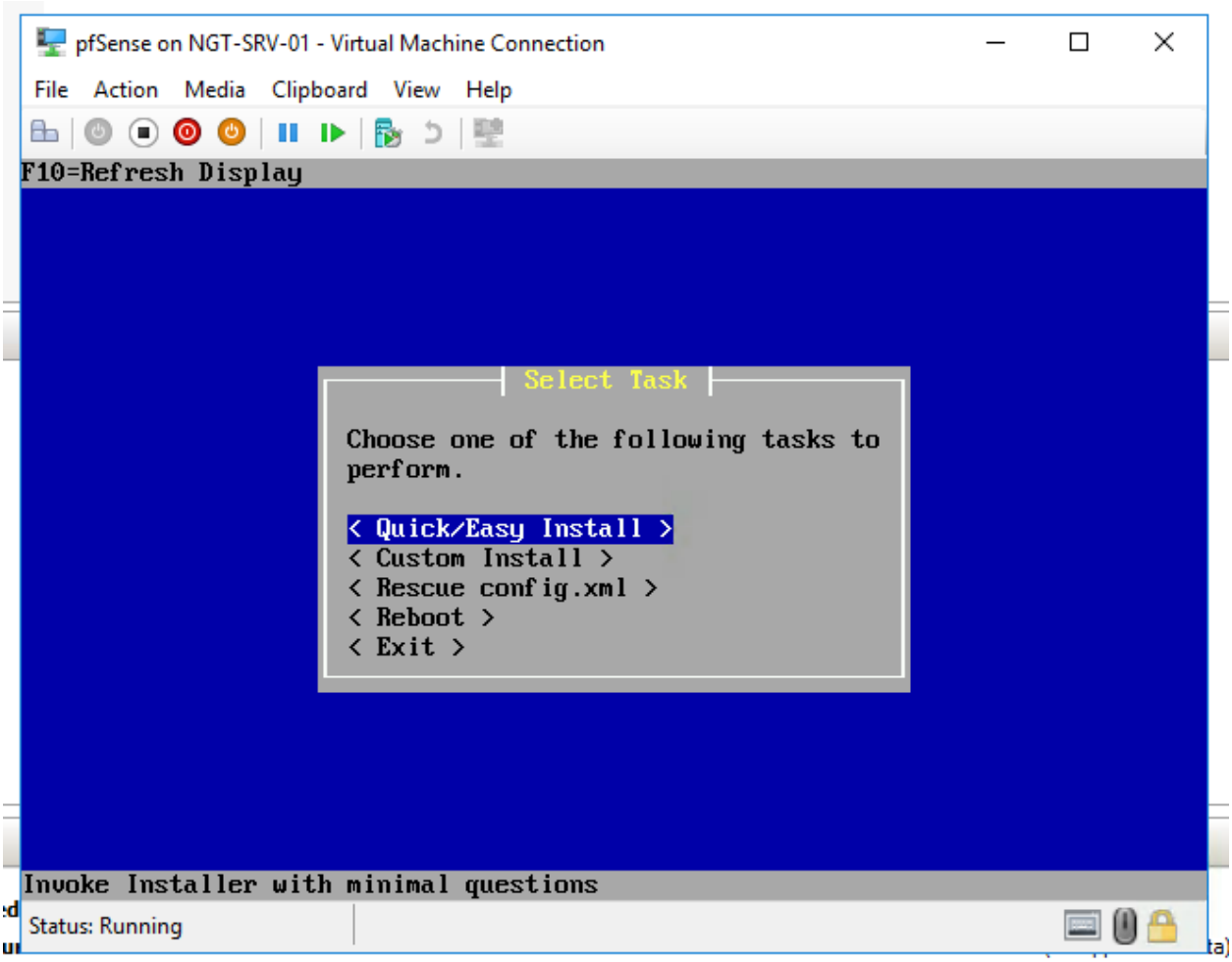
After successfully creating and configuring the pfSense virtual machine, it's time to start it.



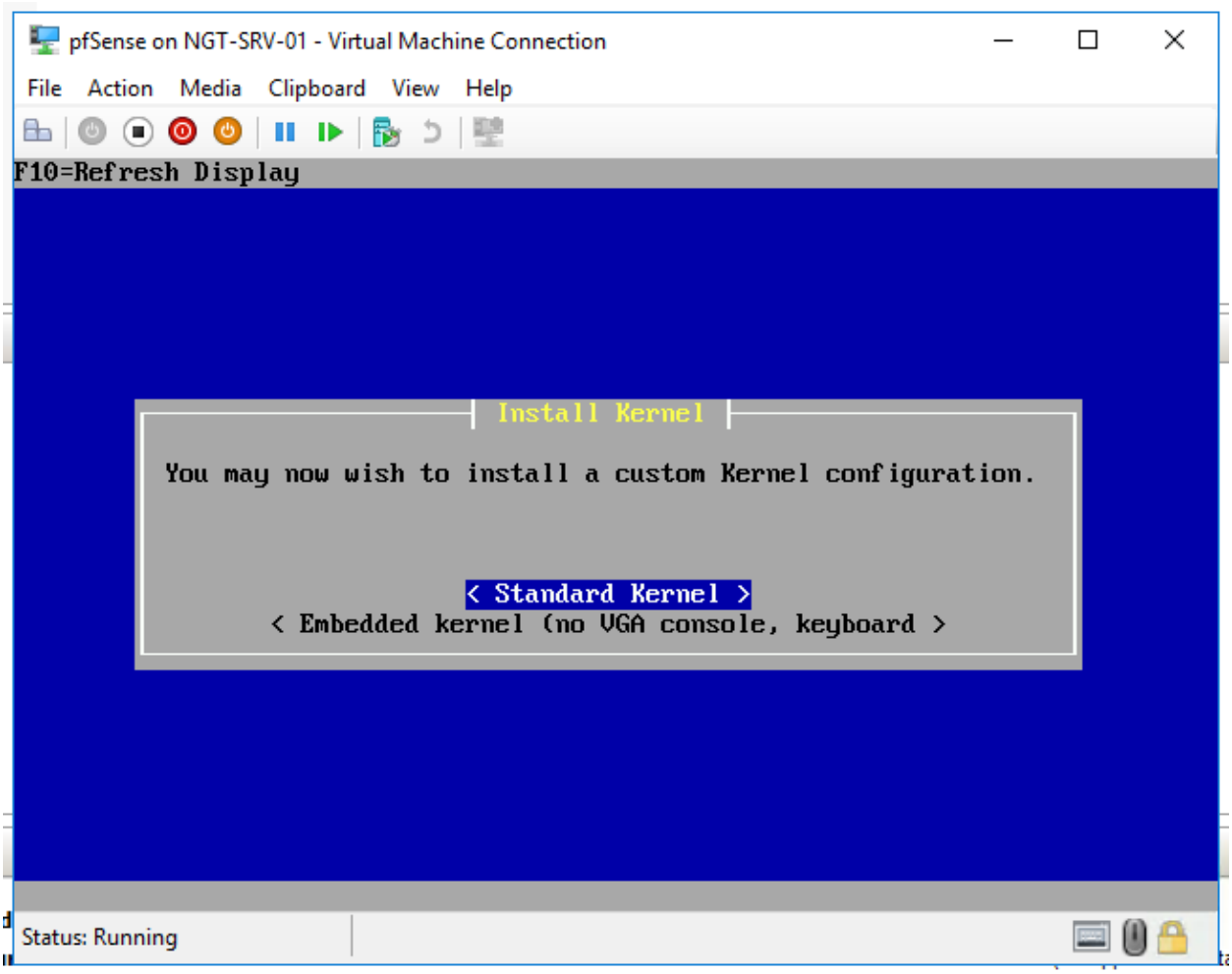
Wait for the virtual machine to boot up and press **I** to invoke installer.



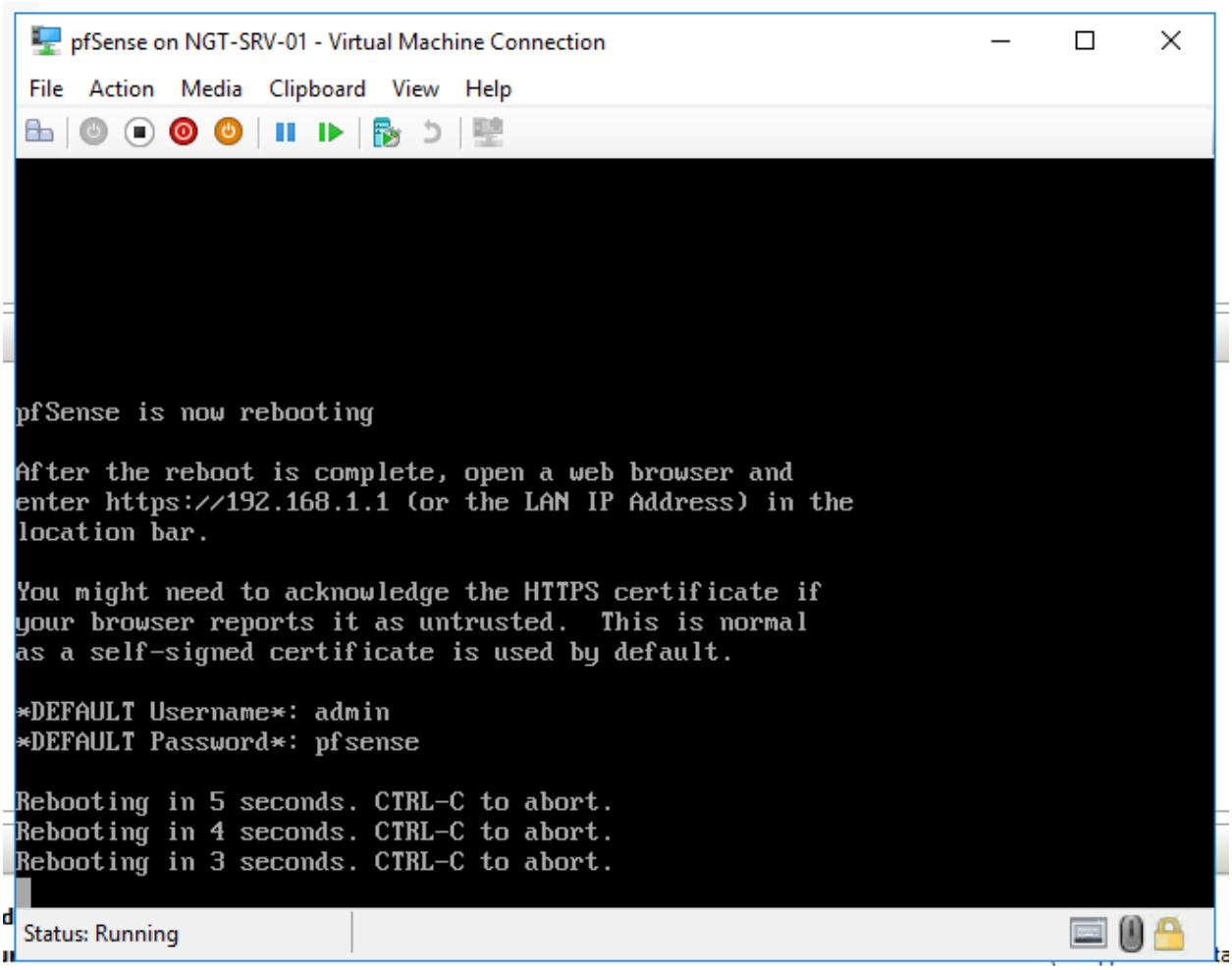
Once installer boots up select the **Quick/Easy Install** and follow the installer steps.



When prompted, select the standard kernel and continue the installation.

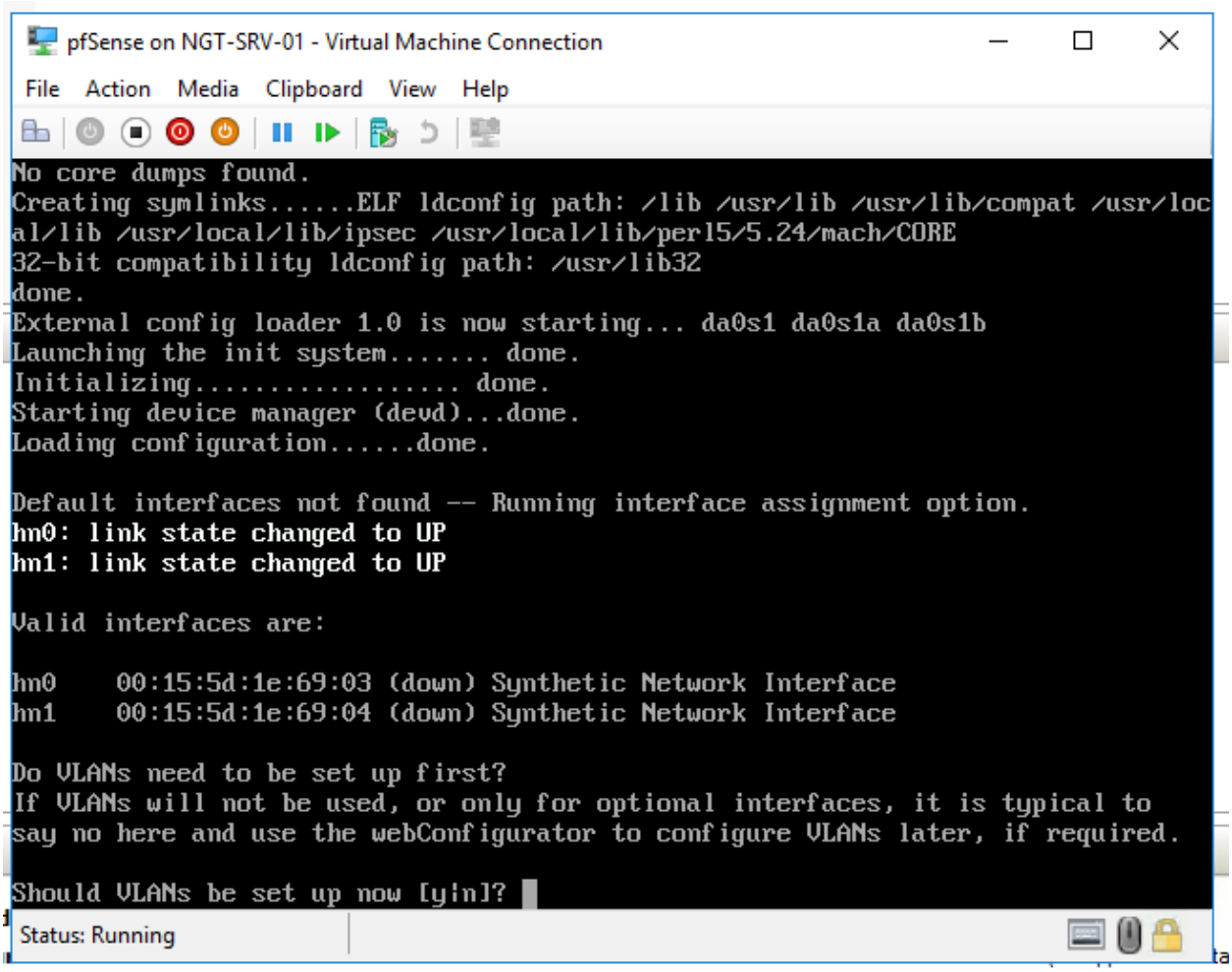


After installation is complete, select reboot and eject the ISO.

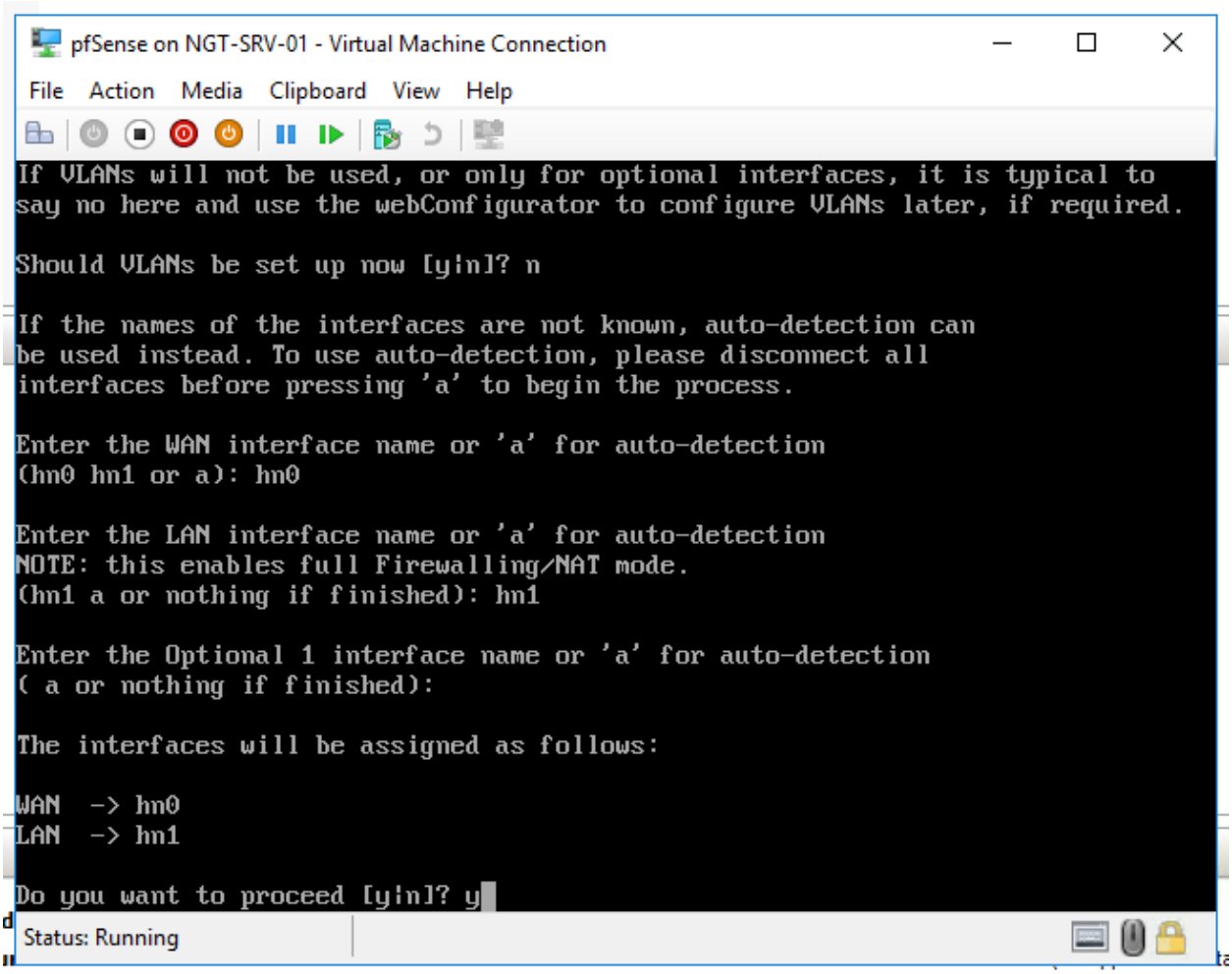


First boot and interfaces assignment

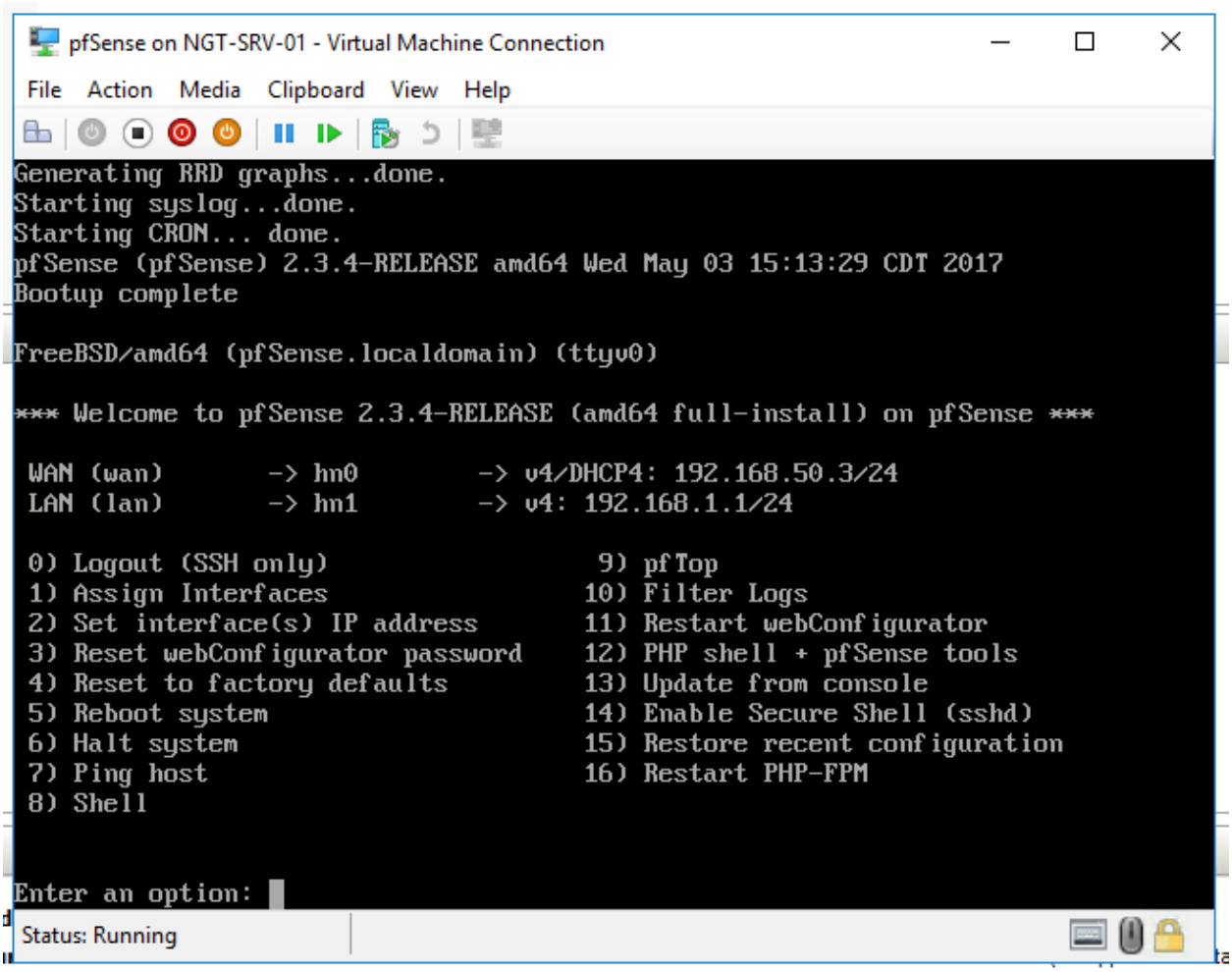
The pfSense virtual machine should boot up quickly and prompt for interface assignments. Select **N** to not set up VLAN's now.



In the following steps assign WAN and LAN interfaces to the appropriate network adapters. The MAC address can be verified against the virtual machine settings.



After assigning interfaces, pfSense software will finish the boot-up. Verify both interfaces have the correct IP addresses.



Congratulations! The virtual machine is now running pfSense software on Microsoft Hyper-V.

Guide under construction, may have minor errors

34.73 Virtualizing with Proxmox® VE

This following article is about building and running pfSense® software on a virtual machine under Proxmox Virtual Environment (VE). The guide also applies to any newer Proxmox VE version. Article covers Proxmox VE networking setup and firewall virtual machine setup process. The guide does not cover how to install Proxmox VE.

A basic, working, virtual machine will exist by the end of this article.

34.73.1 Assumptions

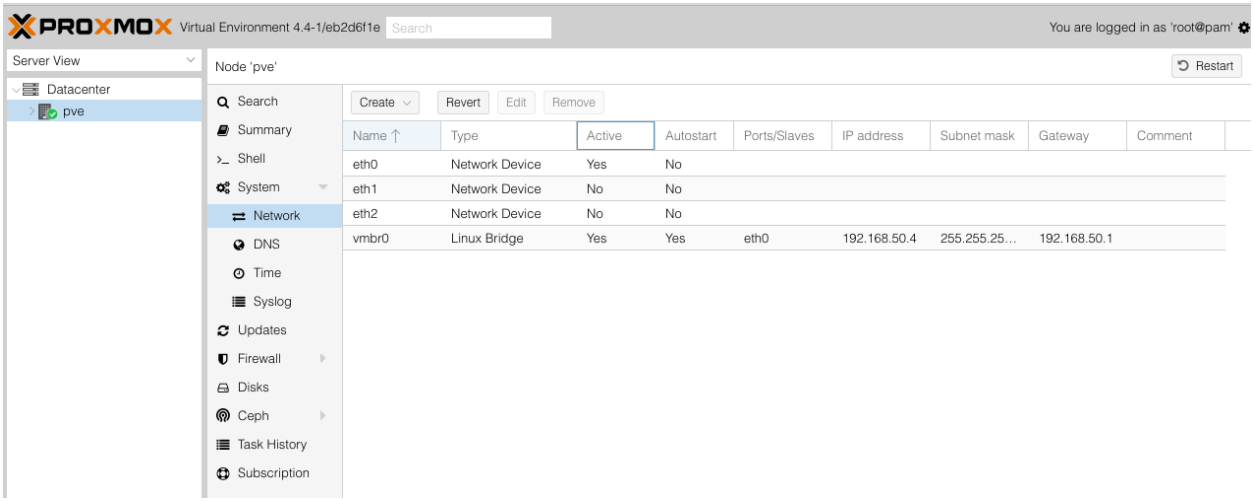
- Proxmox VE host is up and running
- Host has at least two network interfaces available for WAN and LAN.
- pfSense software ISO image is present on the Proxmox VE host

Basic Proxmox VE networking

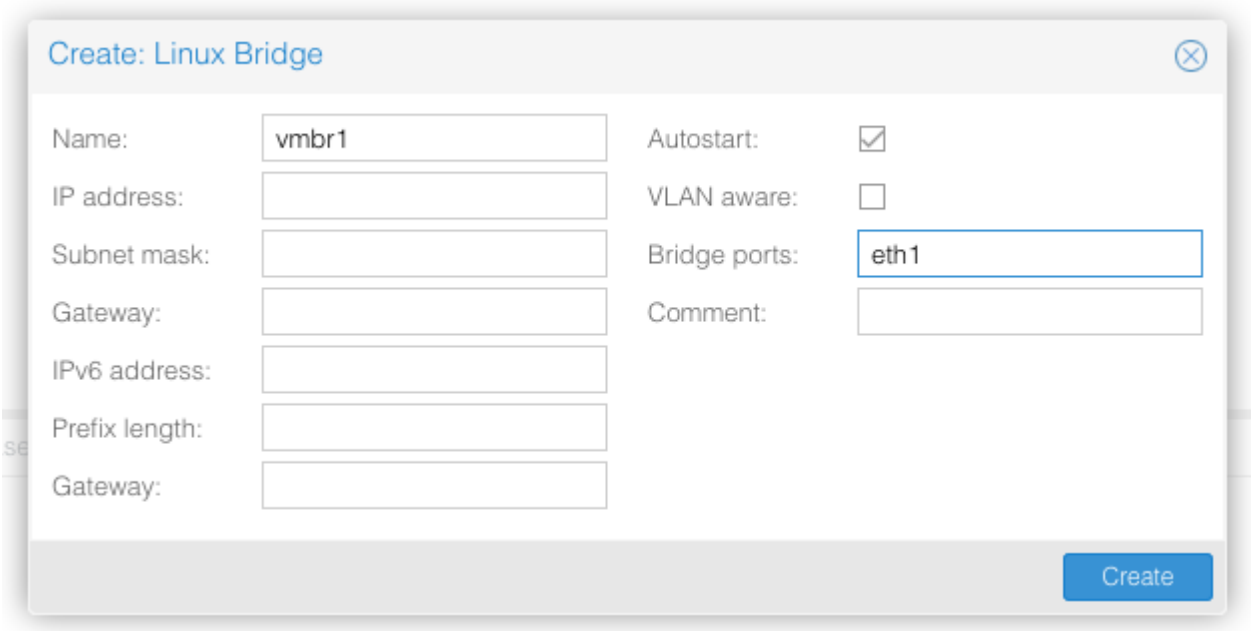
First create two Linux Bridges on Proxmox VE, which will be used for LAN and WAN on the firewall VM.

- Select the host from the server view
- Navigate to **System > Network**

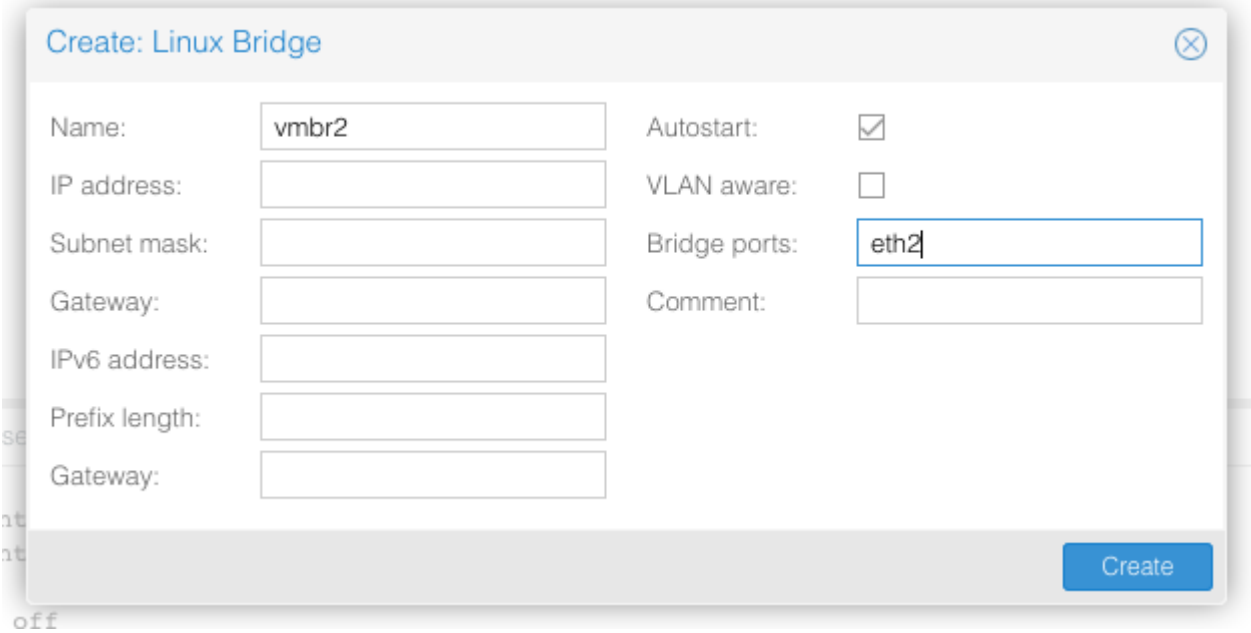
This example uses `eth1` and `eth2` interfaces for the firewall, while `eth0` is for Proxmox VE management.



- Click **create**
- Select Linux Bridge
- Enter `eth1` under **Bridge ports**

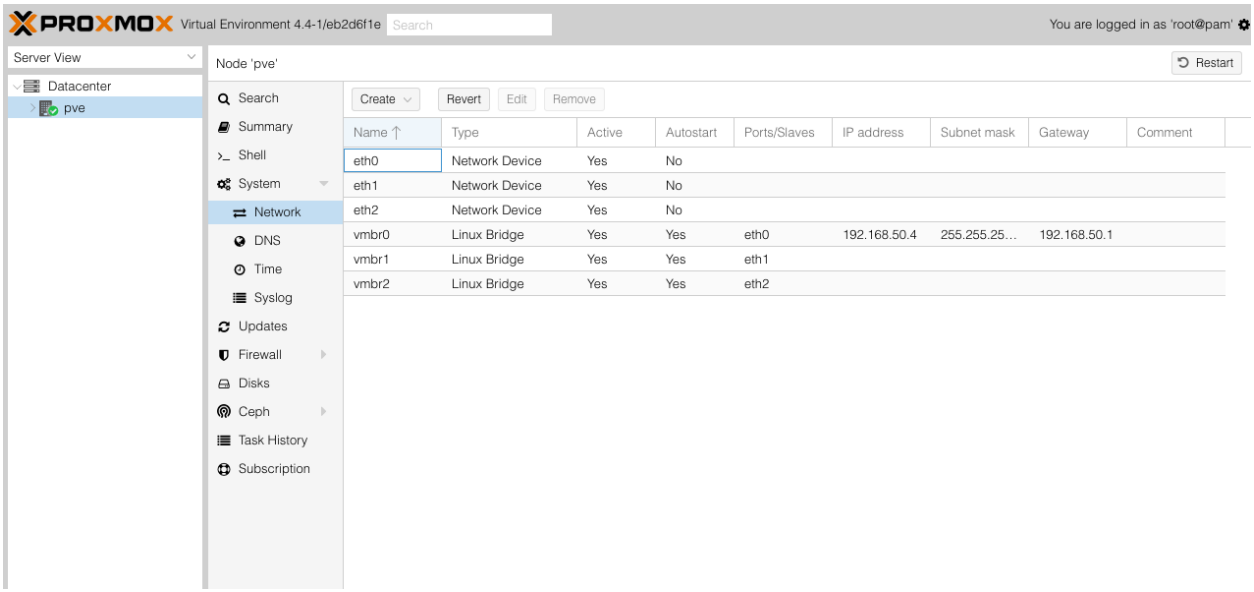


Repeat the process to add another Linux Bridge, this time add `eth2` under **Bridge ports**.



Proxmox VE networking should now display two Linux bridges like on the following screenshot.

Warning: Proxmox VE requires a reboot if the interfaces are not marked **Active**.



Creating a virtual machine

After creating WAN and LAN Linux bridges, now proceed to create a new virtual machine.

- Click **Create VM** from the top right section to display the new virtual machine wizard
- Navigate to the **General** tab
- Enter a **Name** for the VM (e.g. `firewall`)
- Navigate to the **OS** tab
- Set the following options:

  Use CD/DVD disc image file Selected

  Storage *local*

  ISO image Select the previously uploaded ISO image

  Guest OS Type Other
- Navigate to the **System** tab
- Set the following options:

  Graphic card *SPICE*

  > **Note:** The SPICE console uses less CPU when idle and supports more advanced console features than the default console. It is compatible with the VNC Proxmox VE console as well as the more advanced virt-viewer console application.

- Navigate to the **Hard Disk** tab
- Set the following options:

  Bus/Device VirtIO Block

  Disk Size Enter an appropriate disk size, no less than `8` GB.
- Navigate to the **CPU** tab

- Set the following options:

  Socket `1`

  Cores `1` or more cores as needed

  Type **Host** to match the CPU on the hypervisor hardware

- Navigate to the **Memory** tab
- Set the following options:

  Memory At least `1024` MB

  Minimum Memory Use the same value as **Memory**

- Navigate to the **Network** tab
- Set the following options:

  Bridge *vmbr1*

  Model *VirtIO (paravirtualized)*

- Navigate to the **Confirm** tab
- Review the settings and make any final corrections if necessary
- Click **Finish**
- Wait for the VM creation process to finish

Now add another network adapter to the VM:

- Expand the **Server View** list on the left to show the contents under **Datacenter** and the name of this hypervisor node (e.g. **pve**, **proxmox**, etc.)
- Select the newly created virtual machine from list
- Click **Hardware** in the right pane
- Click **Add**
- Click **Network Device**
- Set the following options:

  Bridge *vmbr2*

  Model *VirtIO (paravirtualized)*

- Click **Add**

Review the hardware list for the VM and confirm it now contains two network interfaces.

Starting and configuring the virtual machine

After creating a new virtual machine and adding network interfaces, it is time to start the virtual machine.

- Expand the **Server View** list on the left to show the contents under **Datacenter** and the name of this hypervisor node (e.g. **pve**, **proxmox**, etc.)
- Select the newly created virtual machine from list
- Click **Start**

- Click **Console** on the left, under **Summary**

> **Note:** The **Console** button at the top will launch the console in a new window, which depending on the settings may require an additional client installation such as virt-viewer.

When the VM starts it will boot into the installer automatically. From there, follow the installation steps as usual, and reboot when finished.

After the virtual machine reboots, the console will stop at an interfaces assignment prompt.

- Type `n` and press `Enter` to skip VLAN configuration
- Enter `vtnet0` for WAN
- Enter `vtnet1` for LAN
- Press `Enter`
- Type `y` and press `Enter` to complete the interface assignment

After interfaces have been assigned, the VM will complete the boot process.

Disable Hardware Checksums with Proxmox VE VirtIO

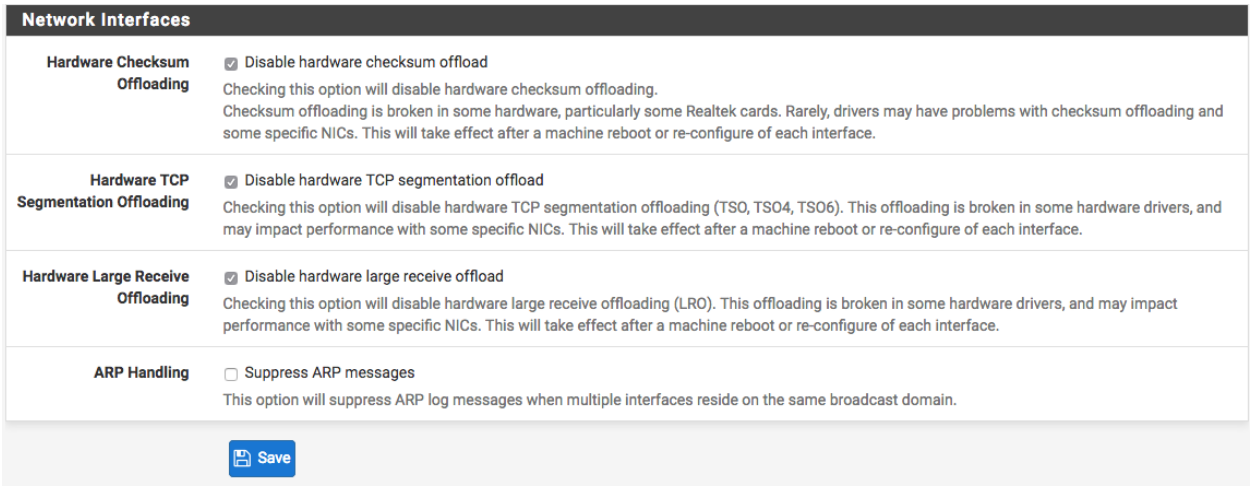
When using VirtIO interfaces in Proxmox VE, hardware checksums **must** be disabled.

> **Warning:** Do not skip this step, otherwise the virtual machine will not properly pass traffic. Accessing the firewall may be sluggish at first, but changing this setting will correct that as well.

After the installation and interfaces assignment processes are complete, connect to the assigned LAN port from another computer.

To disable hardware checksum offload:

- Navigate to **System > Advanced**, **Networking** tab
- Locate the **Networking Interfaces** section
- Check **Disable hardware checksum offload**
- Click Save
- Reboot the firewall from **Diagnostics > Reboot** or the console menu



Congratulations, the virtual machine installation and configuration on Proxmox VE is now complete.

34.74 General

- *Basic Firewall Configuration Example*
- *Blocking Web Sites*
- *Using an External Wireless Access Point*
- *Using Software from FreeBSD*
- *Using NAT and FTP without a Proxy*
- *Configuring pfSense Software for Online Gaming*
- *Migrating an Assigned LAN to LAGG*
- *Accessing a CPE/Modem from Inside the Firewall*
- *Exporting NetFlow with softflowd*
- *Configuring Switches with VLANs*
- *Using the Shaper Wizard to Configure ALTQ Traffic Shaping*

34.75 DNS

- *Configuring DNS over TLS*
- *Configuring BIND as an RFC 2136 Dynamic DNS Server*
- *Blocking External Client DNS Queries*
- *Redirecting Client DNS Requests*

34.76 Authentication

- *Accessing the Firewall Filesystem with SCP*
- *Granting Users Access to SSH*
- *External User Authentication Examples*
- *Authenticating Users with Google Cloud Identity*
- *Using EAP and PEAP with FreeRADIUS*
- *Using Mobile One-Time Passwords with FreeRADIUS*
- *Authenticating from Active Directory using RADIUS/NPS*

34.77 Firewall/NAT

- *Allowing Remote Access to the GUI*
- *Preventing RFC1918 Traffic from Exiting a WAN Interface*
- *Accessing Port Forwards from Local Networks*
- *Configuring NAT for a VoIP PBX*
- *Configuring NAT for VoIP Phones*

34.78 Routing

- *Dynamic Routing Protocol Basics*
- *Configuring IPv6 Through A Tunnel Broker Service*
- *Configuring Multi-WAN for IPv6*
- *Routing Public IP Addresses*

34.79 High Availability

- *High Availability Configuration Example*
- *High Availability Configuration Example with Multi-WAN*
- *High Availability Configuration Example without NAT*

34.80 IPsec

- *IPsec Site-to-Site VPN Example with Pre-Shared Keys*
- *IPsec Site-to-Site VPN Example with Certificate Authentication*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-MSCHAPv2*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-RADIUS*
- *IPsec Remote Access VPN Example Using IKEv2 with EAP-TLS*
- *Configuring IPsec IKEv2 Remote Access VPN Clients*
- *IPsec Remote Access VPN Example Using IKEv1 with Xauth*
- *IPsec Remote Access VPN Example Using IKEv1 with Pre-Shared Keys*
- *Routing Internet Traffic Through a Site-to-Site IPsec Tunnel*
- *Connecting to Cisco PIX/ASA Devices with IPsec*
- *Connecting to Cisco IOS Devices with IPsec*

34.81 L2TP/IPsec

- *L2TP/IPsec Remote Access VPN Configuration Example*
- *Connecting to L2TP/IPsec from Android*

34.82 OpenVPN

- *OpenVPN Site-to-Site Configuration Example with SSL/TLS*
- *OpenVPN Site-to-Site Configuration Example with Shared Key*
- *OpenVPN Remote Access Configuration Example*
- *Adding OpenVPN Remote Access Users*
- *Installing OpenVPN Remote Access Clients*
- *Configuring a Single Multi-Purpose OpenVPN Instance*
- *Authenticating OpenVPN Users with FreeRADIUS*
- *Authenticating OpenVPN Users with RADIUS via Active Directory*
- *Connecting to an OpenVPN Access Server*
- *Connecting OpenVPN Sites with Conflicting IP Subnets*
- *Routing Internet Traffic Through A Site-To-Site OpenVPN Tunnel*
- *Bridging OpenVPN Connections to Local Networks*

34.83 WireGuard

- *WireGuard Remote Access VPN Configuration Example*
- *WireGuard Site-to-Site VPN Configuration Example*
- *WireGuard VPN Client Configuration Example*

34.84 Cache/Proxy

- *A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid*
- *Authenticating Squid Package Users with FreeRADIUS*
- *Configuring the Squid Package as a Transparent HTTP Proxy*
- *Setting up WPAD Autoconfigure for the Squid Package*

34.85 Virtualization

- *Virtualizing pfSense with VMware vSphere / ESXi*
- *Virtualizing pfSense with Hyper-V*
- *Virtualizing with Proxmox® VE*

CHAPTER

THIRTYFIVE

MENU GUIDE

35.1 System

The **System** menu contains choices for the firewall itself, general and advanced options, updates, add-on packages, users, and routing.

Advanced Advanced settings for the firewall, hardware, SSH, notifications, tunables, and many others. See *Advanced Configuration Options*.

Cert Manager Manage Certificate Authorities, Certificates, and Certificate Revocation Lists (x.509). See *Certificate Management*.

General Setup General settings such as hostname, domain, and DNS servers. See *General Configuration Options*.

High Avail. Sync Controls how pfSense® nodes in a High Availability (HA) cluster synchronize states and configuration. See *High Availability*.

Logout Logs out of the GUI, returning the user back to the login screen. See *User Management and Authentication*.

Package Manager Additional software add-ons for pfSense to expand its functionality. See *Packages*.

Routing Manage gateways, static routes, and gateway groups for multi-WAN. See *Routing*.

Setup wizard The Setup Wizard performs the basic initial configuration. See *Setup Wizard*.

Update Upgrade pfSense to the latest version. See *Upgrading using the WebGUI*.

User Manager Manage users, groups, and authentication servers (RADIUS or LDAP) for GUI access, VPN access, etc. See *User Management and Authentication*.

Note: If a user has the *WebCfg - System: User Password Manager* privilege, this menu option leads that user to a page where they can change their own password but not make changes to other users.

User Settings If per-user settings are enabled, this page provides a way for users to override default behavior options found under **General Setup**.

35.2 Interfaces

The **Interfaces** menu contains an entry for assigning interfaces, and entries for each currently assigned interface. Assigned interfaces will show their configured names, or the standard names if they have not been changed (e.g. WAN, LAN, OPTx)

Assignments Assign interfaces to logical roles (e.g. WAN, LAN, OPT), and create/maintain VLANs and other types of virtual interfaces. See *Interface Configuration*, *Interface Types and Configuration*, and *Virtual LANs (VLANs)*.

WAN Configure the WAN interface. See *Interface Configuration*.

LAN Configure the LAN interface. See *Interface Configuration*.

OPTx Configure any additional optional interfaces. See *Interface Configuration*.

35.3 Firewall

The **Firewall** menu entries configure firewall rules, NAT rules, and their supporting structure.

Aliases Manages collections of IP addresses, networks, or ports to simplify rule creation and management. See *Aliases*.

NAT Manages NAT rules that control port forwards, 1:1 NAT, and Outbound NAT behavior. See *Network Address Translation*.

Rules Configures firewall rules. There is one tab on this screen for each configured interface, plus tabs for groups and different VPN types, when enabled. See *Introduction to the Firewall Rules screen*.

Schedules Manages time-based rule schedules. See *Time Based Rules*.

Traffic Shaper Manages traffic shaping/Quality of Service (QoS) settings. See *Traffic Shaper*.

Virtual IPs Configure Virtual IP addresses which enable pfSense® to handle traffic for more than one IP address per interface, typically for NAT rules or High Availability. See *Virtual IP Addresses*.

35.4 Services

The **Services** menu contains items which control services provided by daemons running on the firewall. See *Services*.

Captive portal Controls the Captive Portal service which directs users to a web page for authentication before permitting Internet access. See *Captive Portal*.

DHCP relay Configures the DHCP relay service which proxies DHCP requests from one network segment to another. See *DHCPv4 & DHCPv6 Relay*.

DHCP server Configures the DHCP service which provides automatic IP address configuration for clients. See *DHCPv4 Server*.

DHCPv6 Relay Configures the DHCP relay service for IPv6 which proxies DHCPv6 requests from one network segment to another. See *DHCPv4 & DHCPv6 Relay*.

DHCPv6 Server & RA Configures the DHCP service for IPv6 and Router Advertisements which provide automatic IPv6 address configuration for clients. See *DHCPv6 Server*.

DNS Forwarder Configures the built-in caching DNS forwarder. See *DNS Forwarder*.

DNS Resolver Configures the built-in caching DNS resolver. See *DNS Resolver*.

Dynamic DNS Configures Dynamic DNS services ("dyndns") which updates a remote name server when the WAN IP address of this firewall has changed. See *Dynamic DNS*.

IGMP Proxy Configures the Interior Group Management Protocol proxy for passing multicast traffic between interfaces. See *IGMP Proxy*.

NTP Configures the Network Time Protocol server daemon. See *NTPD*.

PPPoE Server Configures the PPPoE server which accepts and authenticates connections from PPPoE clients on local networks. See *PPPoE Server*.

SNMP Configures the Simple Network Management Protocol (SNMP) daemon to allow network-based collection of statistics from this firewall. See *SNMP*.

UPnP & NAT-PMP Configures the Universal Plug and Play (UPnP) & NAT Port Mapping Protocol service which automatically configures NAT and firewall rules for devices which support the UPnP or NAT-PMP standards. This menu entry only appears if more than one interface is assigned. See *UPnP & NAT-PMP*.

Wake on LAN Configures Wake on LAN entries which remotely wake up local client devices. See *Wake on LAN*.

35.5 VPN

The **VPN** menu contains items pertaining to Virtual Private Networks (VPNs), including IPsec, OpenVPN and L2TP. See *Virtual Private Networks*.

IPsec Configure IPsec VPN tunnels, mobile IPsec, and IPsec settings. See *IPsec*.

L2TP Configure L2TP services and users. See *L2TP VPN*.

OpenVPN Configure OpenVPN servers and clients, as well as client-specific configuration. See *OpenVPN*.

35.6 Status

The **Status** menu entries display status information and logs for various system components and services.

Captive Portal When Captive Portal is enabled, this entry shows user and voucher status. See *Captive Portal*.

CARP (failover) Shows the status of CARP IP addresses on this firewall, such as MASTER/BACKUP state for each CARP VIP. Also has controls for HA maintenance mode. See *Check CARP status*.

Dashboard A shortcut back to the main page of the pfSense® firewall, which displays general system information. See *Dashboard*.

DHCP leases Shows a list of all IPv4 DHCP leases assigned by this firewall and provides controls based on those leases, such as adding static mappings. See *Leases*.

DHCPv6 leases Shows a list of all IPv6 DHCP leases assigned by this firewall. See *Leases*.

Filter Reload Shows the status of the last filter reload request, including active reload actions. Also provides a means to force a filter reload, and to force an XMLRPC configuration sync when HA is configured. See *Troubleshooting Firewall Rules*.

Gateways Shows the status of gateways, and gateway groups for multi-WAN. See *Routing*.

Interfaces Shows the hardware status for network interfaces, equivalent to using `ifconfig` on the console. See *Interface Status*.

IPsec Shows the status of any configured IPsec tunnels. See *IPsec*.

Monitoring Shows graphed data for system statistics such as bandwidth used, CPU usage, firewall states, etc. See *Monitoring Graphs*.

NTP Shows the status of the Network Time Protocol server daemon. See *NTPD*.

OpenVPN Shows the status of any configured OpenVPN instances. See *Checking the Status of OpenVPN Clients and Servers*.

Package logs View logs from certain supported packages.

Queues Shows the status of traffic shaping queues. See *Monitoring the Queues*.

Services Shows the status of system and package service daemons. See *Service Status*.

System logs Shows logs from the system and system services such as the firewall, DHCP, VPNs, etc. See *System Logs*.

Traffic graph Displays a dynamic realtime traffic graph for an interface. See *Traffic Graphs*.

UPnP & NAT-PMP Shows a list of any currently active UPnP port forwards. This entry is only present when the firewall contains more than one interface. See *UPnP & NAT-PMP*.

Wireless Shows a list of any currently available wireless networks in range, along with signal levels. This menu entry is only present if the firewall has an assigned wireless interface. See *Check Wireless Status*.

35.7 Diagnostics

Items under the **Diagnostics** menu perform various diagnostic and administrative tasks.

ARP Table Displays a list of devices as seen locally by the firewall. The list includes an IP address, MAC address, Hostname, the Interface where the device was seen, and other related information.

Authentication Tests authentication to a defined RADIUS or LDAP server. See *Troubleshooting Authentication*.

Backup & Restore Backup and restore configuration files. See *Backup and Recovery*.

Command Prompt Execute shell commands or PHP code, and upload/download files to/from the firewall. **Use with caution**.

DNS Lookup Executes a DNS lookup to resolve hostnames for diagnostic purposes, and to test connectivity to DNS servers. See *DNS Lookup*.

Edit File Edit a file on the firewall filesystem.

Factory defaults Resets the configuration back to default. Be aware, however, that this does not alter the filesystem or uninstall package files; it only changes configuration settings. See *4) Reset to factory defaults*.

GEOM Mirrors If the firewall contains a GEOM disk mirror, this page shows the status of the mirror and provides controls for managing the mirror.

Halt system Shuts down the firewall and turns off the power where possible. See *6) Halt system*.

Limiter Info Shows the status of any Limiters and the traffic flowing inside them. See *Checking Limiter Usage*.

NDP Table Shows a list of local IPv6 devices as seen by the firewall. The list includes an IPv6 address, MAC address, hostname (if known to the firewall), and the interface.

Packet Capture Perform a packet capture to inspect traffic, and then view or download the results. See *Packet Captures from the WebGUI*.

pfInfo Displays statistics about the packet filter, including general traffic rates, connection rates, state table info, and various other counters. See *pfInfo*.

pfTop Displays a list of the top active connections by a selectable metric such as bytes, rate, age, etc. See *Viewing States with pfTop*.

Ping Sends ICMP echo requests to a given IP address, sent via a chosen interface.

Reboot system Reboots the firewall. This can take several minute to complete, depending on the hardware and enabled features. See *5) Reboot system*.

Routes Shows the contents of the routing table. See *Viewing Routes*.

SMART Status Displays diagnostic information about disk drives, if supported by the hardware. Can also run drive tests. See *S.M.A.R.T. Hard Disk Status*.

Sockets Displays a list of processes on the firewall that are bound to network ports, listening for connections or making connections outbound from the firewall itself.

States Shows the currently active firewall states. See *Firewall States*.

States Summary Displays information about the state table, to see activity summarized by IP address. See *States Summary*.

System Activity Shows memory usage and a list of active processes and system threads on the firewall, the output is from `top -aSH`. See *System Activity (Top)*.

Tables Displays and edits the contents of various firewall tables and aliases. See *Viewing the Contents of Tables*.

Test Port Performs a simple TCP connection test from the firewall to determine if a remote host is accepting connections on a specified port.

Traceroute Trace the route taken by packets between this firewall and a remote system. See *Using traceroute*.

This section is a guide to the standard menu choices available in pfSense® software. This guide will help to quickly identify the purpose of a given menu option, and refer to places in the documentation where those options are discussed in further detail.

Packages can add items to any menu, so check each menu or consult the documentation for a package to locate its menu entries. Typically, packages install entries under the **Services** menu, but there are numerous exceptions.

CHAPTER

THIRTYSIX

GLOSSARY OF TERMS

DNS An acronym for Domain Name System.

ICMP An acronym for Internet Control Message Protocol.

HTTP An acronym for Hypertext Transfer Protocol.

IP An acronym for Internet Protocol.

LAN An acronym for Local Area Network.

NAT An acronym for Network Address Translation.

SSH An acronym for Secure Shell.

TCP An acronym for Transmission Control Protocol.

UDP An acronym for User Datagram Protocol.

WAN An acronym for Wide Area Network.

VM An acronym for Virtual Machine.

VPN An acronym for Virtual Private Network.

Note: Though technically abbreviations read letter by letter are initialisms, not acronyms, this document refers to both as acronyms to be more accessible to readers.

CHAPTER

THIRTYSEVEN

DEVELOPMENT

These articles cover advanced topics related to developing on or with pfSense® software.

37.1 General Development Information

37.1.1 Software Release Schedule

Releases are made when they are ready. A public schedule is not available at this time, but release announcements and progress messages are made on the Netgate Blog.

We provide maintenance releases as needed, typically a couple per year. These include primarily bug fixes and security updates.

To follow the progress of a release, visit the pfSense Redmine.

37.1.2 Reporting Issues with pfSense Software

This page serves as a guide for providing legitimate bug reports with pfSense® software. Most of the bug reports from outside users are not bugs at all, simply misconfigurations. Or alternatively, do not contain nearly enough information for a developer to replicate the problem. If a bug report does not contain the appropriate information to verify a legitimate bug, it will be rejected.

Attention: The pfSense Redmine site is not a discussion platform and is never to be used for support issues.

Unless a precise and specific description of the problem is noted, including which portion(s) of the underlying system are causing problems/misconfigured and how, first inquire about the problem using one of our available support resources. If/when a specific bug is found, a bug ticket can be opened.

An example of an invalid bug report is "XYZ doesn't work!" without appropriate accompanying detail. An alternative, to make that a legitimate bug report, is "XYZ doesn't work, because the underlying xyz.conf has option1=1 where it should be option1=5 when I check box A on thispage.php in the web interface".

See also:

How to report bugs effectively.

Where to submit

For anything that is not a confirmed, specific, detailed bug report, post to the Netgate Forum or pfSense subreddit first. Those sites are platforms where the problem can be discussed and the specifics required to replicate the issue can be identified.

After discussion and confirmation of a specific, legitimate bug report on the forum or pfSense subreddit, please open a ticket in the pfSense Redmine.

Attention: The pfSense Redmine site is not a discussion platform and is never to be used for support issues.

What to Include

When submitting a bug report to the pfSense Redmine site, fill out the report completely and include enough supporting information to reproduce the issue.

Use the following guidelines when completing the issue report.

File issues with the base system under pfSense and issues with packages belong under pfSense Packages.

Note: Not all fields will be available to all users.

Tracker Use the appropriate issue tracker type, following these guidelines:

Bug Problems, unexpected behavior, crashes, or when the other categories do not apply.

Feature Feature requests or changes in (working) behavior.

Todo Tasks or work that need to be completed that are not specifically bugs or features.

Subject A brief but complete and accurate description of the problem. If the problem is specific to one page or file, prefix the subject with that page filename.

Example: `services_rfc2136_edit.php: Save and Force Update button does not perform any action`

Description A full description of the problem. Include any relevant detail, supporting evidence such as log entries, and if possible a complete recount of how to reproduce the bug that includes every necessary step.

Warning: Information in this issue tracker is public! Do not include any personal information such as usernames, e-mail addresses, IP addresses, and so on. If developers require that information they will request it privately.

- Please use appropriate formatting, such as <pre> tags around log data or command output.
- Files may also be attached, such as logs or crash dumps. Check for personal data before attaching files and obfuscate/mask info as needed.

Status Leave this as *New*.

Priority Leave this as *Normal* or set lower for minor issues. **Higher priorities will be set by developers only when appropriate**.

Assignee Leave this **empty** unless directed by a developer to assign it to them directly.

Category Pick the closest relevant category for the issue, if possible, or a generic category such as "Web Interface" for GUI issues and "Operating System" for OS-level issues.

Target Version Leave this **empty** unless directed by a developer to assign a specific target. **Developers will assign a target version after evaluating the issue.**

Affected Version Pick the version number of the firewall experiencing the issue.

Affected Architecture Pick *All* unless the issue is known to only affect a single architecture.

Leave other fields blank.

37.1.3 Obtaining Panic Information for Developers

Crash dump functionality is built into every kernel on current versions of pfSense® software. Crash dumps are automatically saved on installations with swap space, or printed to the console on other platforms.

Viewing and Submitting a Crash Dump

After a panic or crash leading to a reboot, the system attempts to recover the contents of the crash dump while booting back up.

If the system was able to read and process the contents of the crash dump, it converts the crash dump into a crash report. The GUI then displays a prompt on the dashboard to view the contents of the crash report.

Note: For privacy reasons, crash reports cannot be automatically submitted to Netgate for review.

Download the report or copy the contents from the GUI, and review the included data to ensure there is no private or identifiable information inside. If the data is OK, create a post on the Netgate Forum with the contents of the crash report. Attaching the crash report archive file(s) from the GUI is the best practice as that yields the most information in the easiest format to read.

Crash Dump Format

The crash dumps are in FreeBSD textdump format and held in `/var/crash` after they have been recovered.

Crash dump archives are named `textdump.tar.<n>` where `<n>` starts at `0` and is incremented if older archives are still present in `/var/crash`.

Crash dump archives contain the following files:

`config.txt` The kernel configuration file

`ddb.txt` The output from the debugger scripts run during the crash dump

`msgbuf.txt` The contents of the kernel message buffer (`dmesg`)

`panic.txt` The panic message, if available. Typically a more detailed version of this is found at the end of `msgbuf.txt`.

`version.txt` The kernel version string

Serial Console Crash Dump

On systems with a serial console, connect to the serial console and record the scrollback buffer when a crash happens, or there may not be a way to retrieve the crash dump otherwise.

Install without Swap Space

If the system is running an installation without swap space, it may not be able to take a crash dump or automatically restart, and may stop at a `db>` prompt on the console. Capture the output there, and also the output of the `bt` command at that prompt, then manually restart the system.

37.1.4 FreeBSD Issue Policy

The pfSense® team relies on the work provided by the FreeBSD project, as the base operating system of pfSense software. On occasion, pfSense users will run into problems with FreeBSD that are beyond our ability to help with. This page provides some basic guidance on what to do in these scenarios.

Most frequently these issues are driver or hardware-specific bugs. We do not have any developers who work on drivers in FreeBSD, and cannot assist with such issues.

When driver problems are encountered, we suggest installing a stock FreeBSD release on the hardware, replicating the problem, and reporting it to the appropriate FreeBSD list. This is the only way the problem will be resolved and even at that will only be resolved in future releases. Alternatively, use different hardware.

If a kernel panic is encountered, we may be able to get someone to analyze the backtrace. Feel free to post one to the mailing list or forum. See: *Obtaining Panic Information for Developers*

The only exception to this is for issues that affect all or a large number of pfSense users. Examples of this could be problems with PF, CARP, IPsec, or any number of other components. We will track and work to resolve issues of this nature, but cannot possibly coordinate testing and fixes for one off issues encountered by users.

37.1.5 Requesting New pfSense Features

To request new pfSense® features, first, submit a feature request on the pfSense Redmine. If it is approved, write the functionality and submit a pull request to the applicable pfSense repositories.

37.1.6 System Patches Package

The System Patches package allows patches to be added, either from the official code repository or ones pasted in from e-mail or other sources.

This makes it easier to test and deploy small changes instead of pulling in many changes.

Installing the package

As with any other pfSense® package, it's available via the package repository.

- Navigate to **System > Packages**, **Available Packages** tab.
- Find **System Patches** in the list.
- Click  at the end of its row, then confirm, to install.

Patches may now be managed at **System > Patches**.

Adding a patch

- Go to **System > Patches**.
- **Read the text and warnings!**.
- Click  to add a new patch.
- Enter a **Description** to identify the patch.
- Enter any one of:
  - Commit ID (e.g. `4573641589d50718b544b778cea864cfd725078a`) in the **URL/Commit ID** field.
  - GitHub commit URL (e.g. `https://github.com/pfsense/pfsense/commit/4573641589d50718b544b778cea864cfd725078a`) in the **URL/Commit ID** field.
  - GitHub Pull Request (PR) URL with '.diff' appended, such as `https://github.com/pfsense/pfsense/pull/XXXX.diff` where XXXX is the PR number. Set Path Strip = 2 if it does not adjust automatically.
  - Full URL to a patch from another source (e.g. `https://redmine.pfsense.org/attachments/594/0001-Add-support-for-aliases-in-DNS-Forwarder-fixes-2410.patch`) in the **URL/Commit ID** field.
  - Leave **URL/Commit ID** blank and paste the contents of a patch into **Patch Contents** text area.
- Set the **Path Strip Count**. GitHub commit IDs and URLs should be count of `2` (fixed automatically on save). Patches from other sources will need to be set appropriately.
  - If a path like **a/src/etc/inc/filter.inc** is in the patch header, strip off the **a/src** so a strip count of `2` is needed. If it's deeper, such as **home/me/patches/etc/inc/filter.inc**, strip `3` levels in that example.
- A **Base Directory** of `/` is assumed for the patches, but an alternate base may be applied if a patch does not supply a full path to the file it is patching (e.g. `/usr/local/www`).
- Set the **Ignore Whitespace** option appropriately. Patches from GitHub should work with either whitespace setting, patches from other sources may need the option set to ignore whitespace, especially if they contain DOS line endings rather than UNIX.
- click **Save**.

Applying/Reverting a patch

- If a URL or commit ID was entered, there will be a **fetch** link. Click **fetch** and the patch will be retrieved only.
- Once a patch has been fetched (or pasted in) so its contents are present, a set of new links is shown: **Test**, **Apply**, **Revert**.
  - Only options which would succeed are shown. For example, if a patch will apply, **Apply** will appear. Same for **Revert**. **Test** will always show, and will test an apply and revert to tell if either one would work.
- To apply the patch, simply click **Apply** and it will apply the patch. The available link for the patch will then change to say **Revert** instead. To revert, click **Revert**.

Troubleshooting

- Click **Re-Fetch** for remote patches to make sure a clean copy of the patch is present.
- Click **Test** to run a test and then click **Detail** next to either the apply or revert line to get the full patch output
- If the above test output mentions **No file to patch**, double check the **Path Strip Count** and/or the **Base Directory**.
- If every part of a patch fails, try toggling **Ignore Whitespace**.

Automatic Apply

When adding a patch, if **Auto-Apply Patch** is checked then the patch will be automatically applied during each package sync/boot.

The patches may be reordered in the list to arrange them so they apply in a specific order automatically, in case one patch depends on a previous patch.

Known issues

See also:

The pfSense bug tracker contains a list of known issues with this package.

Package Support

This package is currently supported by Netgate TAC to those with an active support subscription.

37.1.7 Using the pfSense PHP Shell

Using the PHP pfSense® shell allows configuration of the config.xml file directly without needing to use the webGUI. Using this system can also allow rapid deployment of pfSense software and/or the setup of exotic configurations.

The following will show an example session, with the text coming from the “help” command in the PHP shell.

Follow each line or group of lines to run with “exec;”:

```
*** Welcome to pfSense ***

 WAN (wan)       -> vmx0       -> v4/DHCP4: 198.51.100.3/24
                                  v6/DHCP6: 2001:db8::ffff:22d6/128
 LAN (lan)       -> vmx1       -> v4: 10.3.0.1/24
                                  v6/t6: 2001:db8:1:eee0:20c:29ff:fe45:260/60

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Disable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option: 12


Starting the pfSense developer shell....

Welcome to the pfSense developer shell

Type "help" to show common usage scenarios.

Available playback commands:
     changepassword disablecarp disabledhcpd disablerefererchecк enableallowallwan␣
↪enablecarp
     enablesshd externalconfiglocator generateguicert gitsync installpkg listpkg␣
↪removepkgconfig
     removeshaper restartdhcpd restartipsec svc uninstallpkg

pfSense shell: help

     Enter a series of commands and then execute the set with "exec".

     For example:
     echo "foo"; // php command
     echo "foo2"; // php command
     ! echo "heh" # shell command
     exec

     Example commands:

     record <recordingfilename>
     stoprecording
     showrecordings

     parse_config(true);  # reloads the $config array

     $temp = print_r($config, true);
     more($temp);

     /* to output a configuration array */
     print_r($config);
```

(continues on next page)

(continued from previous page)

```
/* to output the interfaces configuration portion of config.xml */
print_r($config['interfaces']);

/* to output the dhcp server configuration */
print_r($config['dhcpd']);

/* to exit the  developer shell */
exit

/* to output supported wireless modes for an interface */
print_r(get_wireless_modes(\"ath0\"));

/* to enable SSH */
$config['system']['enablesshd'] = true;

/* change OPTX to the OPT interface name such as BACKHAUL */
$config['interfaces']['optx']['wireless']['standard'] = "11a";
$config['interfaces']['optx']['wireless']['mode'] = "hostap";
$config['interfaces']['optx']['wireless']['channel'] = "6";

/* to enable dhcp server for an optx interface */
$config['dhcpd']['optx']['enable'] = true;
$config['dhcpd']['optx']['range']['from'] = "192.168.31.100";
$config['dhcpd']['optx']['range']['to'] = "192.168.31.150";

/* to disable the firewall filter */
$config['system']['disablefilter'] = true;

/* to enable an interface and configure it as a DHCP client */
$config['interfaces']['optx']['disabled'] = false;
$config['interfaces']['optx']['ipaddr'] = "dhcp";

/* to enable an interface and set a static IPv4 address */
$config['interfaces']['wan']['enable'] = true;
$config['interfaces']['wan']['ipaddr'] = "192.168.100.1";
$config['interfaces']['wan']['subnet'] = "24";

/* to save out the new configuration (config.xml) */
write_config();

/* to reboot the system after saving */
system_reboot_sync();
```

Recording and Playback

For example check out this sessions which automates a number of commands. After typing those sets of commands in 5+ times it gets old quick. Record and playback to the rescue.

Recording a session

```
# /usr/local/sbin/pfSsh.php

Starting the pfSense developer shell....

Welcome to the pfSense developer shell

Type "help" to show common usage scenarios.

Available playback commands:
     changepassword disablecarp disabledhcpd disablereferercheck enableallowallwan␣
↪enablecarp
     enablesshd externalconfiglocator generateguicert gitsync installpkg listpkg␣
↪removepkgconfig
     removeshaper restartdhcpd restartipsec svc uninstallpkg

pfSense shell: record resetrrd
Recording of resetrrd started.
pfSense shell: require_once("filter.inc");
pfSense shell: require("shaper.inc");
pfSense shell: require_once("rrd.inc");
pfSense shell: ! rm /var/db/rrd/*.rrd
pfSense shell: enable_rrd_graphing();
pfSense shell: setup_gateways_monitor();
pfSense shell: stoprecording
Recording stopped.
pfSense shell: exit
```

Playing back a session

```
# /usr/local/sbin/pfSsh.php
Starting the pfSense developer shell....

Welcome to the pfSense developer shell

Type "help" to show common usage scenarios.

Available playback commands:
     changepassword disablecarp disabledhcpd disablereferercheck enableallowallwan␣
↪enablecarp
     enablesshd externalconfiglocator generateguicert gitsync installpkg listpkg␣
↪removepkgconfig
     removeshaper resetrrd restartdhcpd restartipsec svc uninstallpkg

pfSense shell: playback resetrrd

Playback of file resetrrd started.

pfSense shell: exit
```

Sessions can be played back directly from the command line as well:

```
# pfSsh.php playback resetrrd
```

37.2 pfSense Software Development

37.2.1 Getting Started with pfSense Software Development

There is no single specific starting point for joining the pfSense® development effort, but the following items are helpful in getting started:

- Review the *Developer Style Guide*
- Become familiar with the pfSense git repositories (and github in general):
  - Current repositories used for developing pfSense software and its dependencies:
    - pfSense - The main source repository for pfSense software, containing the GUI code, builder code, and related scripts.
    - FreeBSD-src - The OS source code used to build pfSense software
    - FreeBSD-ports - Build information for supporting software used in pfSense software, and code for some of our custom programs/daemons/modules and packages
  - Older repositories which are not used for current/new development:
    - pfSense-tools - Deprecated repository with old build tools used for pfSense software version 2.2.x and before
    - pfSense-packages - Repository with package metadata and files used for pfSense software version 2.2.x and before
    - xmlrpc-server - Repository with the deprecated XMLRPC server used for packages on pfSense software version 2.2.x and before
- Review the list of open bug reports and other issues.
- Submit changes as pull requests on github.

Our developers will review the submissions, offer feedback, and merge the changes if they are acceptable.

37.2.2 Referencing Tickets in Commit Messages

By placing a special keyword in a commit message, the bug tracking system (Currently Redmine) can associate a commit with a specific ticket automatically, creating a link in the ticket to the relevant commits.

When using these keywords immediately follow them by a # and then the ticket number, such as “Ticket #1234”. They are not case sensitive.

The following keywords will reference a ticket but take no action on the ticket status:

- refs, references, IssueID, ticket, bug, feature, todo, redmine

The following keywords will not only reference the ticket, but automatically move the ticket to a feedback state:

- fix, fixes, fixed, close, closes, closed, resolve, resolves, resolved, implement, implements, implemented, finish, finishes, finished

Keep this in mind when submitting changes in github pull requests for existing issues.

37.2.3 Submitting a Pull Request via GitHub

Submitting a Pull Request (PR) via GitHub is the fastest and best way to contribute source code changes to the pfSense® project.

Using a PR allows developers to easily review and comment on changes, allows easy testing via patches from the *System Patches* Package, and allows the changes to be easily merged into the project.

Creating a PR is a relatively straightforward process but we do have a few guidelines and suggestions to follow when submitting contributions:

- Ensure the bug or feature has an entry on https://redmine.pfsense.org – *Create a new entry* if one does not exist
  - The only exceptions to this are for very minor typo/wording fixes or for FreeBSD-ports package updates that have already been given an OK/green light
- Read through and follow the *Developer Style Guide* when making changes to the source code
- Read through the GitHub documentation on pull requests
- Create a fork of the correct repository (e.g. pfSense/pfSense for the base system, or pfSense/FreeBSD-ports for packages)
- Always work on the master branch of the pfSense/pfSense repository or the devel branch of pfSense/FreeBSD-ports
  - The exception to this is when making PRs for multiple branches due to syntax or other differences which require variations that prevent a regular cherry-pick or similar from copying the commit to other branches
- When making commits to the fork, reference the relevant Redmine entry as mentioned in *Referencing Tickets in Commit Messages*
- When ready to submit the changes, create a pull request from the fork to the appropriate branch
  - The PR title should be a short summary of the changes and include a reference to the relevant Redmine issue number(s)
  - The PR description should include a longer explanation of the proposed changes and link to the relevant Redmine issue number(s)
- After submitting the pull request, add a link to the PR on the related Redmine issue so the entries can be cross-referenced

37.2.4 Checking the Current FreeBSD Version

The *Versions of pfSense and FreeBSD* article contains a table of the versions used in various releases of pfSense® software and FreeBSD, or follow these instructions:

From the Dashboard in the webGUI, look at the **System Information** widget. It is displayed under the **Version** section. Clicking the FreeBSD version displays the full kernel *uname -a* output.

In the *SSH console* or *Execute Shell Command* field in the GUI, run the following:

```
uname -mrs
```

37.2.5 Creating Dashboard Widgets

Getting Started

Creating widgets is simple. First, create the html code to be displayed, save it to a file named widget_name.widget.php, and put it into the /usr/local/www/widgets/widgets directory on the firewall.

Do not include any **<body>**, **<html>**, or pgtitle definitions, etc. Just the basic HTML code for what needs to be displayed. See the current widgets for examples.

The file must be named in the name_name.widget.php format. No spaces are allowed. The name that will displayed is the name of the file. For example the Traffic Graphs widget file is named traffic_graphs.widget.php.

And that's it! The rest of the widget (buttons, border, dragging, sequence, etc) is handled automatically by the dashboard.

To include custom PHP or JavaScript code into the widget upon rendering, create a file named widget_name.inc for PHP or widget_name.js for JavaScript. Copy .inc files in the /usr/local/www/widgets/include directory and JavaScript files into the /usr/local/www/widgets/javascript directory.

Saving Data

To save configuration data for the widget, some more work is required.

To store the data to the XML config use the following code:

```
<input type="hidden" id="widget_name-config" name="widget_name-config" value="">
```

where widget_name is the name of the widget file minus the .widget.php. An example is:

```
<input type="hidden" id="traffic_graphs-config" name="traffic_graphs-config" value="">
```

Then copy the data into the value field of this hidden input. The value stored to this input field the data will be stored as a string into the config.

To retrieve the stored value for the widget use variables such as $config['widgets']['widget_name_setting'], where widget_name is the name of the widget. An example would be $config['widgets']['filterlogentriesinterfaces']

For even more control, provide specific inputs for the user to modify. Use the two code sections below to give more control over the widget.

To show the configuration button use the following code:

```
<script type="text/javascript">
//<![CDATA[
    selectIntLink = "widget_name-configure";
    textlink = document.getElementById(selectIntLink);
    textlink.style.display = "inline";
//]]>
</script>
```

where widget_name is the name of the widget file minus the .widget.php. An example is selectIntLink = "traffic_graphs-configure";

To show additional configuration options use a section at the top of the widget. Use the following code and insert whatever is needed between the and tags:

```
<div id="services_status-settings" class="widgetconfigdiv" style="display:none;">
  <form action="/widgets/widgets/services_status.widget.php" method="post" name=
↪"iformd">
    Comma separated list of services to NOT display in the widget<br />
    <input type="text" size="30" name="servicestatusfilter" class="formfld unknown"␣
↪id="servicestatusfilter" value="<?= $config['widgets']['servicestatusfilter'] ?>" />
    <input id="submitd" name="submitd" type="submit" class="formbtn" value="Save" />
  </form>
</div>
```

Note that it is necessary to copy the data from the various input fields to the correct backend settings:

```
if(isset($_POST['servicestatusfilter'])) {
  $config['widgets']['servicestatusfilter'] = htmlspecialchars($_POST[
↪'servicestatusfilter'], ENT_QUOTES | ENT_HTML401);
  write_config("Saved Service Status Filter via Dashboard");
  header("Location: /index.php");
}
```

Customizing the Title and Linking to page

By default the name of the widget file is what is shown on the Widget in the dashboard. This title can be changed and also have a link to another page inserted.

To configure the widget to use a certain name other than the name of the file, create an .inc file and insert that file into the /usr/local/www/widgets/include directory.

For example if a widget is named abc.widget.php, the include file will be abc.inc

In this .inc file use the following code:

```
<?php //set variable for custom title
$abc_title = "A B C custom";
$abc_title_link = "abc.php";
?>
```

An example of this can be taken from the interfaces.inc file:

```
<?php //set variable for custom title
$interfaces_title = "Interfaces";
$interfaces_title_link = "interfaces_assign.php";
?>
```

37.2.6 Enabling Additional PHP Modules

In some cases, such as packages or advanced Captive Portal code, additional PHP extensions may be required that are not enabled.

There are several PHP extensions that are included in the pfSense® binary distribution on pfSense software version 2.2 and before, but are left disabled by default to conserve resources since they are not required by the base system. These disabled modules may be activated using the “dynamodules” system.

The extensions included and activated vary by pfSense software version, look in */usr/local/lib/php/* to see which extensions are present, and check the output of *php -m* to see what is enabled on a firewall already.

Examples of extensions that are included but not activated may include:

- bz2
- mysql
- pdo_sqlite
- sockets
- sysvmsg
- sysvsem
- sysvshm
- tokenizer
- xdebug

To activate one of these, create a file named for the extension in the */etc/dynamodules* directory, and then trigger a rewrite of the *php.ini* file.

Activate the *mysql.so* module:

```
$ mkdir -p /etc/php_dynamodules/
$ touch /etc/php_dynamodules/mysql
$ /etc/rc.php_ini_setup
$ php -m | grep mysql
mysql
```

A reload of the WebGUI process may also be required. Use option **11** from the console/ssh menu. On pfSense software version 2.2 and later, also use option **16** to restart PHP-FPM. A reboot would also fully activate the module, but should not be necessary.

If a module is needed that is not included nor activated, a copy of the module may be obtained from an equivalent version FreeBSD system using a matching PHP version. Drop it into the correct lib directory, and activate it with dynamodules. This is a lot riskier, and may be prone to break in various ways, thus is it not supported nor recommended.

Installing Additional Modules on pfSense Software version 2.3

Due to the modular nature of pfSense software version 2.3 using pkg, the extra modules that were included in the base system before are no longer there by default. They may be easily added using *pkg install*.

For example, to install the *mysqli* extension, use:

```
pkg install php56-mysqli
```

If the package name is not known, use search to find it:

```
pkg search mysql
```

37.2.7 Executing Commands at Boot

There are three primary options for executing custom commands at boot time: shellcmd, earlyshellcmd, and shell script.

The shellcmd package can manage the shellcmd and earlyshellcmd tags in the GUI, so *config.xml* values need not be edited by hand.

At boot time, the earlyshellcmd commands are executed first, shellcmd is executed later in the boot process, and the shell scripts are executed at the very end when packages are initialized.

shellcmd option

The hidden **config.xml** option **<shellcmd>** will run the command specified towards the end of the boot process.

To add a shellcmd to a configuration, either use the shellcmd package or edit the config by hand. To edit the config, back it up via **Diagnostics > Backup/restore**, and open the resulting XML file in a text editor (other than the stock Windows Notepad). Above the **</system>** line, add a line such as the following:

```
<shellcmd>mycommand -a -b -c 123</shellcmd>
```

Where `mycommand -a -b -c 123` is the command to run. Multiple lines may be added to execute multiple commands. Save the changes and restore the modified configuration.

earlyshellcmd option

The hidden **config.xml** option **<earlyshellcmd>** will run the command specified at the beginning of the boot process. Normally, **<shellcmd>** should be used instead, though this may be necessary in some circumstances. Similarly to **<shellcmd>**, to add a **<earlyshellcmd>** option, either use the shellcmd package or edit it in by hand. To edit it manually, backup the configuration, open it in a text editor, and add a line such as the following above **</system>**:

```
<earlyshellcmd>mycommand -a -b -c 123</earlyshellcmd>
```

Where `mycommand -a -b -c 123` is the command to run. Multiple **<earlyshellcmd>** lines may be added to execute multiple commands. Save the changes and restore the modified configuration.

Shell script option

Any shell script can be placed in the **/usr/local/etc/rc.d/** directory. The filename must end in **.sh** and it must be marked as executable (`chmod +x myscript.sh`). Every shell script ending in **.sh** in this directory will be executed at boot time.

The first two options are preferable as they are retained in the config file and hence do not require additional modifications should the storage medium be replaced and reinstalled, or if the configuration is restored to a different piece of hardware.

37.2.8 Using a Debug Kernel

pfSense® software version 2.3 has debug symbols for the kernel and modules that can be added to aid in debugging. To install the debug symbols, use the following command:

```
pkg install pfSense-kernel-debug
```

37.2.9 Using gitsync to Update pfSense Between Snapshots

Most often, upgrading to a new snapshot is the best way to get updated code when tracking a prerelease version of pfSense® software (alpha, beta, RC, etc).

However, since it does take some time to generate new snapshots, one can often get by with pulling new code from the pfSense git repository instead of reloading a whole new snapshot. This process is known as a gitsync.

Warning: This syncs only PHP changes, without any binary changes. At times, PHP changes require associated binary changes that only come from an upgrade using a snapshot or other upgrade image. **Unless development is followed closely and the ramifications of all changes are understood, or unless system breakage is not a concern, do not use this!** For most users, this action should only be taken if directed to do so by a developer.

This only works with code or files that are not compiled. For example: PHP Code, configuration files, Web GUI pages, etc.

There are two ways to perform a gitsync, both perform the same function:

Method 1: From the console menu, press option 12 to start a developer shell, then type:

```
> playback gitsync master
```

Method 2: From a normal shell (console menu option 8), type the following command:

```
# pfSsh.php playback gitsync master
```

The "master" part of the command tells the gitsync process to grab the code for the "master" branch, a.k.a. HEAD, which as of this writing is the development code for 2.3. That can be replaced with RELENG_2_2 for 2.2.x.

For example, on 2.2.4, to sync post-release code changes for 2.2.x, use:

```
# pfSsh.php playback gitsync RELENG_2_2
```

Troubleshooting

The gitsync script will attempt to install git automatically. However, if an error occurs that git was not found and a gitsync has never been performed before, the git package may need to be added manually. This can be done from the GUI using the 'Available Packages' list, or from a shell prompt:

```
pfSsh.php playback installpkg git
```

Git URL Moved

If gitsync was run prior to the pfSense repository moving to github, and the default 2.0 URL was used, one of the following must be performed before the next gitsync: 1:

```
rm -rf /root/pfsense/
```

2:

```
cd /root/pfsense/pfSenseGITREPO/pfSenseGITREPO; git remote set-url origin git://
↪github.com/pfsense/pfsense.git
```

Warning: There have been times where a gitsync is not enough to bring a system up to date and could leave the system in a broken state. Read the commit logs from after the last snapshot update until the most current code to be sure.

37.2.10 What are HEAD, RELENG 2 1, RELENG 2 0, etc

This is the terminology used to refer to a specific git branch of pfSense® source code.

HEAD (master)

HEAD, also known as -HEAD or "master", refers to the development version of pfSense software, where all new features are first added. When a release nears, HEAD is branched to a RELENG (Release Engineering) branch. This follows the FreeBSD project's development model.

RELENG_2_3

This branch is under active development for the 2.3.1 release.

RELENG_2_3_0

This branch is under active maintenance for 2.3 errata.

Others

For links to other branches, past and present, see *Versions of pfSense and FreeBSD*

See also:

- *Developer Style Guide*

37.3 pfSense Package Development

37.3.1 Developing Packages

For developers familiar with FreeBSD, pkgng, and PHP, packages are not difficult to create. End users and organizations can benefit from developing a package that does not exist.

To submit a new package, create a pull request on Github for the pfSense FreeBSD-Ports Repository so the work can be evaluated for inclusion into the package system for everyone to see.

When developing packages, always target the latest development version of pfSense software first.

pfSense Package System

On pfSense software, every pfSense package is also a FreeBSD port. These are installed and managed via `pkg`, even when using the GUI to add or remove packages. Binary packages from FreeBSD are added as dependencies of the pfSense package, so they are installed automatically as well, along with any any of their required dependencies.

The basic idea is to make packages for pfSense software similar to FreeBSD packages, but with customizations. One way this is achieved is by adding metadata about packages to the firewall configuration when it is installed, and also creating the configuration screen of an application using XML. pfSense software provides an optional framework to create the web interface and to store it in the XML configuration file of the firewall. The package writer is expected to convert the data from XML to the native format of the application.

For help converting older style packages to the new Bootstrap GUI, see *Converting Packages to Bootstrap*.

Package System

pfSense packages are typically composed of:

- A Manifest File
- Package configuration file(s)
- Supporting files (`.inc` files, additional `.php` web interface files, etc.)

See also:

See *Package Port Directory Structure* for a more in-depth list of files and their locations in the package structure.

Manifest File

The manifest file is located inside the package's port directory:

```
<category>/pfSense-pkg-<package name>/files/usr/local/share/pfSense-pkg-<package name>
↪/info.xml
```

For example:

```
sysutils/pfSense-pkg-Cron/files/usr/local/share/pfSense-pkg-Cron/info.xml
```

This file contains basic information about the package. The format of the manifest XML file is as follows:

```
<package>
    <name>someprogram</name>
    <internal_name>someprogram</internal_name>
    <pkginfolink>https://forum.netgate.com/</pkginfolink>
    <descr><![CDATA[Some cool program]]></descr>
    <website>http://www.example.org/someprogram</website>
    <category>Services</category>
    <version>0.99</version>
    <status>Beta</status>
    <required_version>2.2</required_version>
    <config_file>https://packages.pfsense.org/packages/config/someprogram/someprogram.
↪xml</config_file>
    <maintainer>me@example.com</maintainer>
    <configurationfile>someprogram.xml</configurationfile>
    <only_for_archs>i386 amd64</only_for_archs>
</package>
```

Package Configuration Files

The manifest specifies a Package Configuration File for the package using the `config_file` tag. The convention is to keep this file inside the `files/usr/local/pkg/` directory inside the port structure for the package.

The easiest way to get a feel for the format is to look at existing packages and how they use these configuration files, how their fields loook, how the code behaves, and so on.

The format is:

```
<?xml version="1.0" encoding="utf-8" ?>
<packagegui>
  <name></name>
  <version></version>
  <title></title>
  <include_file></include_file>
  <backup_file></backup_file>
  <aftersaveredirect></aftersaveredirect>
  <configpath></configpath>
  <menu>
    <name></name>
    <section></section>
    <configfile></configfile>
    <tooltiptext></tooltiptext>
    <url>/pkg.php?xml=package.xml</url>
  </menu>
  <service>
    <name></name>
    <rcfile></rcfile>
    <executable></executable>
  </service>
  <tabs>
    <tab>
      <text></text>
      <url></url>
      <active/>
      <tab_level/>
    </tab>
  </tabs>
```

(continues on next page)

(continued from previous page)

```
  <additional_files_needed>
    <prefix></prefix>
    <chmod></chmod>
    <item></item>
  </additional_files_needed>
  <adddeleteeditpagefields>
    <columnitem>
      <fielddescr></fielddescr>
      <fieldname></fieldname>
    </columnitem>
  </adddeleteeditpagefields>
  <fields>
    <field>
      <fielddescr></fielddescr>
      <fieldname></fieldname>
      <description></description>
      <size></size>
      <type></type>
    </field>
  </fields>
  <custom_php_global_functions><!-- PHP function call --></custom_php_global_
↪functions>
  <custom_php_install_command><!-- PHP function call --></custom_php_install_command>
  <custom_php_deinstall_command><!-- PHP function call --></custom_php_deinstall_
↪command>
  <custom_add_php_command><!-- PHP function call --></custom_add_php_command>
  <custom_add_php_command_late><!-- PHP function call --></custom_add_php_command_
↪late>
  <custom_delete_php_command><!-- PHP function call --></custom_delete_php_command>
  <custom_php_resync_config_command><!-- PHP function call --></custom_php_resync_
↪config_command>
  <start_command><!-- PHP function call --></start_command>
  <process_kill_command><!-- PHP function call --></process_kill_command>
</packagegui>
```

Field types

interfaces_selection Combo/list box with interfaces list:

```
<field>
  <fielddescr>Interface Selection</fielddescr>
  <fieldname>interfaces</fieldname>
  <type>interfaces_selection</type>
  <description>Select interfaces to listen on</description>
  <multiple/><!-- (optional) -->
  <size>10</size><!-- (optional) -->
  <hideinterfaceregex>(wan|loopback)</hideinterfaceregex><!-- (optional) -->
  <showvirtualips/><!-- (optional) -->
  <showips/><!-- (optional) -->
  <showlistenall/><!-- (optional) -->
</field>
```

checkbox Field with text description and a enable/disable checkbox:

```
<field>
  <fielddescr>Enable</fielddescr>
  <fieldname>enable_package</fieldname>
  <type>checkbox</type>
  <description>Select this option to enable this config</description>
</field>
```

input Single line text edit element

```
<field>
  <fielddescr>username</fielddescr>
  <fieldname>username</fieldname>
  <type>input</type>
  <description>Enter package username</description>
</field>
```

password Special input element for passwords, all input will be masked with * symbol on gui but clear text on xml config file:

```
<field>
  <fielddescr>password</fielddescr>
  <fieldname>password</fieldname>
  <type>password</type>
  <description>Enter password</description>
</field>
```

textarea Multi-line text edit element:

```
<field>
  <fielddescr>Custom options</fielddescr>
  <fieldname>custom_options</fieldname>
  <type>textarea</type>
  <description>Paste custom config here</description>
  <encoding>base64</encoding><!-- (optional) -->
</field>
```

select Combo Box with dropdown list items:

```
<field>
  <fielddescr>Proxy server</fielddescr>
  <fieldname>proxy_server</fieldname>
  <description><![CDATA[Select proxy server to read logs from]]></description>
  <type>select</type>
    <options>
    <option><name>Squidguard</name><value>squidguard</value></option>
    <option><name>Squid</name><value>squid</value></option>
    </options>
  <multiple/><!-- (optional) -->
  <size>10</size><!-- (optional) -->
</field>
```

info Information text without any options to select:

```
<field>
  <fielddescr>Additional info</fielddescr>
  <fieldname>just_info</fieldname>
  <type>info</type>
```

(continues on next page)

(continued from previous page)

```
  <description>show info text on package gui</description>
</field>
```

button Additional buttons to take additional actions on packages:

```
<field>
 <fielddescr>Reload config</fielddescr>
 <fieldname>reload</fieldname>
 <type>button</type>
 <description>click to force a config reload</description>
 <placeonbottom/><!-- Use this option to place the button besides save default␣
↪button -->
</field>
```

In package `.inc` file, to check what button was selected, use:

```
if (($_POST['Submit'] == 'Save') {...}
if (($_POST['Submit'] == 'Reload') || !isset($_POST['Submit'])){..}
```

Field groups

rowhelper Used in `pkg_edit.php` to add multiple config lines like a table on package gui. Inside rowhelper, add any field type described above:

```
<field>
<fielddescr><![CDATA[Lists]]></fielddescr>
<fieldname>none</fieldname>
<description><![CDATA['Format' - Choose the file format that url will retrieve or␣
↪local file format.]]></description>
<type>rowhelper</type>
  <rowhelper>
    <rowhelperfield>
    <fielddescr>Format</fielddescr>
    <fieldname>format</fieldname>
    <type>select</type>
      <options>
      <option><name>gz</name><value>gz</value></option>
      <option><name>txt</name><value>txt</value></option>
     </options>
    </rowhelperfield>
    <rowhelperfield>
      <fielddescr>Url or localfile</fielddescr>
      <fieldname>url</fieldname>
      <type>input</type>
      <size>75</size>
    </rowhelperfield>
  </rowhelper>
</field>
```

adddeleteeditpagefields Used with `pkg.php` to have multiple config of the same xml page. Useful to access lists, users lists, multi daemon configs, etc:

```
<adddeleteeditpagefields>
  <columnitem>
```

(continues on next page)

(continued from previous page)

```
    <fielddescr>Alias</fielddescr>
    <fieldname>aliasname</fieldname>
  </columnitem>
  <columnitem>
    <fielddescr>Description</fielddescr>
    <fieldname>description</fieldname>
  </columnitem>
  <columnitem>
    <fielddescr>Action</fielddescr>
    <fieldname>action</fieldname>
  </columnitem>
  <columnitem>
    <fielddescr>Update Frequency</fielddescr>
    <fieldname>cron</fieldname>
  </columnitem>
</adddeleteeditpagefields>
```

Binaries from FreeBSD

The actual binaries are normal FreeBSD package binaries for that particular program. Once listed as a dependency for a pfSense package in its *Makefile*, they are built automatically on the pfSense pkg server. There is no need to specify these in XML.

Updating Packages

When updating a package is it important to bump the version in its *Makefile* otherwise the package will not be rebuilt and made available to others.

Repository Branches

When submitting changes, they are typically submitted to the `devel` branch of the FreeBSD-Ports Repository under pfSense. In order to show to all users, the changes must be placed in the current release branch as well, such as RELENG_2_5_2.

Ideally, the changes should be submitted to the development branches and tested on systems pulling packages from the development repository. Once the changes have been tested, they can be placed into the release branch for deployment to everyone.

Testing/Building Individual Packages

For `pkg` files, the files may be copied to the firewall and added with `pkg` directly. The good thing about using `pkg` is that the GUI packages and CLI packages are all the same. Files for the pfSense package are all kept together inside the archive, dependencies such as FreeBSD packages are in separate archives.

The package may be compiled on a local FreeBSD 12.2-STABLE@f4d0bc6aa6b builder, then `pkg delete` the old version and then `pkg add` the new one or use any other `pkg` operations needed.

For example, a basic package like `Cron` is `pfSense-pkg-Cron-0.3.3`, so if a new copy is built and put on the firewall:

```
pkg add /path/to/file/pfSense-pkg-Cron-0.3.3.txz
```

It will also work with `pkg add` and a URL to an `http` or `https` web server.

The process for making a package is:

- Check out the FreeBSD-Ports Repository.
- *Locate the port directory*
- Make changes
- Run `make package`:

```
$ git clone git@github.com:pfsense/FreeBSD-ports.git pfSense-ports
$ cd pfSense-ports/blah/pfSense-pkg-foo/
[hack, hack, hack]
$ make package  (might need sudo)
$ scp work/pkg/pfSense-pkg-foo* root@myfirewall:.
```

And then on the firewall:

```
# pkg add pfSense-pkg-foo-<version>.txz
```

Poudriere could also be setup for a custom repository but in most cases that will be overkill.

There are additional considerations if when adding files, like updating the *plist*, and crafting a new pfSense package from scratch may be tricky if there is no prior knowledge of how the FreeBSD ports tree works, but overall it will be smoother in the long run.

37.3.2 Converting Packages to Bootstrap

For help converting older packages and PHP pages to Bootstrap format, see BOOTSTRAP.md and Bootstrap Conversion Notes for general Bootstrap help and information.

Packages for pfSense® software version 2.3 are built from files in the pfSense copy of the FreeBSD ports repository, not from the older pfSense-packages repository.

pfSense packages contain the files needed for the package to operate and also act as "meta" ports that include the proper dependencies to pull in binary packages if needed. See *Package Port Directory Structure* for more information on the new package port structure. And *Developing Packages* for general information on developing packages.

For package maintainers the primary differences are:

- Which repository to send a PR against
- Locating the package among the other ports can be more difficult. See *Package Port List* for a list to make it easier.
- The package version **must be increased** in order to trigger a new package to be built, which will all happen automatically thanks to poudriere.

The pfSense package port version does not have to be the same as the underlying software version when using a package that requires additional software.

Note: Packages that only use .xml files to draw forms need no conversion, they're OK as-is! Only packages which use PHP files need intervention.

For examples of a conversion, see some work in progress on the OpenVPN Client Export Package and System Patches packages.

The current process should be close to this:

- Fork and check out a copy of the pfSense FreeBSD-ports repository
- Start making changes to the package files in pfsense/freebsd-ports
- Be sure to update the version in the port's Makefile as well as files/usr/local/share/pfSense-pkg-/info.xml and in files/usr/local/pkg/.xml (See *Package Port Directory Structure*)
- Submit a pull request to the pfsense/freebsd-ports repository with the updated files

Once the PR is merged, the package builder will see the version change and automatically build a new copy of the package.

Start a new thread on the Development category on the forum to solicit testing and advice/help as needed.

37.3.3 Package Port Directory Structure

The directory structure of a package for pfSense® software is similar to that of a traditional FreeBSD port.

or more information on working with FreeBSD Ports, see bsd.port.mk.

This page uses the simple Cron package as an example, many other packages are similar. See *FreeBSD Ports Used for Packages* for links to existing packages to copy/clone from.

Category First is the category, which roughly lines up with the category for the package with the caveat that if a pfSense package is based on a FreeBSD port, it should be in the same location (e.g. `squid` is under `www/squid`)

In the case of Cron this is `sysutils`:

```
FreeBSD-ports/sysutils/
```

Main package directory Inside of the category directory, it is always prefixed with `pfSense-pkg-`:

```
FreeBSD-ports/sysutils/pfSense-pkg-Cron/
```

> **Note:** From here on, the `FreeBSD-ports` prefix will be omitted for brevity.

`Makefile` Includes version information, information about binaries, dependencies, install procedures, where to copy files, and so on. Copy an existing one for a similar package and adjust as needed (but do so carefully):

```
sysutils/pfSense-pkg-Cron/Makefile
```

pfSense standard package `install`/`deinstall` scripts

> **Note:** These are the same for all packages. Copy the contents from another existing package.

```
sysutils/pfSense-pkg-Cron/files/pkg-deinstall.in
sysutils/pfSense-pkg-Cron/files/pkg-install.in
```

Description A brief text description of the package:

```
sysutils/pfSense-pkg-Cron/pkg-descr
```

Packing List A list of files installed by the package, for specifics on the format, see the links above:

```
sysutils/pfSense-pkg-Cron/pkg-plist
```

Files Directory The directory where custom files are placed which will be copied to the firewall:

```
sysutils/pfSense-pkg-Cron/files/
```

The structure under the `files/` directory should follow the same conventions as files on the pfSense installation, which typically follows hier(7) from FreeBSD. For example, executable scripts would go under `files/usr/local/bin/`

Privileges An optional file containing privilege information:

```
sysutils/pfSense-pkg-Cron/files/etc/inc/priv/cron.priv.inc
```

Package Code and Configuration The include files and XML files for the package:

```
sysutils/pfSense-pkg-Cron/files/usr/local/pkg/cron.inc
sysutils/pfSense-pkg-Cron/files/usr/local/pkg/cron.xml
```

XML Metadata

```
sysutils/pfSense-pkg-Cron/files/usr/local/share/pfSense-pkg-Cron/info.xml
```

GUI-Accessible Files Files that go into the main directory of the web server:

```
sysutils/pfSense-pkg-Cron/files/usr/local/www/
```

Note: The Cron package uses a directory under /usr/local/www for its files – this is optional.

```
sysutils/pfSense-pkg-Cron/files/usr/local/www/packages/cron/cron.php
sysutils/pfSense-pkg-Cron/files/usr/local/www/packages/cron/cron_edit.php
sysutils/pfSense-pkg-Cron/files/usr/local/www/packages/cron/index.php
```

37.3.4 FreeBSD Ports Used for Packages

The current list of packages in the pfSense® copy of FreeBSD-ports is:

- arping (net/pfSense-pkg-arping)
- AutoConfigBackup (sysutils/pfSense-pkg-AutoConfigBackup)
- Avahi (net/pfSense-pkg-Avahi)
- Backup (sysutils/pfSense-pkg-Backup)
- Cron (sysutils/pfSense-pkg-Cron)
- Darkstat (net-mgmt/pfSense-pkg-darkstat)
- FreeRADIUS2 (net/pfSense-pkg-freeradius2)
- FTP Client Proxy (ftp/pfSense-pkg-FTP_Client_Proxy)

- HAProxy (net/pfSense-pkg-haproxy)
- HAProxy-devel (net/pfSense-pkg-haproxy-devel)
- iftop (net-mgmt/pfSense-pkg-iftop)
- iperf (benchmarks/pfSense-pkg-iperf)
- LADVD (net/pfSense-pkg-LADVD)
- Lightsquid (www/pfSense-pkg-Lightsquid)
- Mail Reports (mail/pfSense-pkg-mailreport)
- MTR (net/pfSense-pkg-mtr-nox11)
- nmap (security/pfSense-pkg-nmap)
- Notes (sysutils/pfSense-pkg-Notes)
- ntopng (net/pfSense-pkg-ntopng)
- Open-VM-Tools (emulators/pfSense-pkg-Open-VM-Tools)
- OpenVPN Client Export (security/pfSense-pkg-openvpn-client-export)
- pfBlockerNG (net/pfSense-pkg-pfBlockerNG)
- RRD Summary (sysutils/pfSense-pkg-RRD_Summary)
- Service Watchdog (sysutils/pfSense-pkg-Service_Watchdog)
- shellcmd (sysutils/pfSense-pkg-Shellcmd)
- siproxd (net/pfSense-pkg-siproxd)
- Snort (security/pfSense-pkg-snort)
- softflowd (net-mgmt/pfSense-pkg-softflowd)
- Squid (3.x) (www/pfSense-pkg-squid)
- SquidGuard (www/pfSense-pkg-squidGuard)
- Sudo (security/pfSense-pkg-sudo)
- Suricata (security/pfSense-pkg-suricata)
- syslog-ng (sysutils/pfSense-pkg-syslog-ng)
- System Patches (sysutils/pfSense-pkg-System_Patches)
- Zabbix-Agent (net-mgmt/pfSense-pkg-zabbix-agent)
- Zabbix-Proxy (net-mgmt/pfSense-pkg-zabbix-proxy)

Note that each package is under a category and prefixed with "pfSense-pkg-".

See *Package Port Directory Structure* for details about the structure of files inside the package directory.

37.3.5 Compiling Software on the Firewall

pfSense® software intentionally does not include a proper environment for compiling software (make, headers/includes, sources, etc) on the installed firewall. Those tools are left out for security and capacity reasons.

A virtual machine or separate system can be setup to compile software, and then the compiled binaries/packages/software can be moved over to the firewall.

When doing this, install a version of FreeBSD that matches up with the version of pfSense software currently in use. A list can be found here: *Versions of pfSense and FreeBSD*

Alternately, install pre-compiled FreeBSD packages as described here: *Installing FreeBSD Packages*

CHAPTER

THIRTYEIGHT

REFERENCES

These documents are for reference purposes and may be about general topics that do not fit into other categories or topics about the documentation itself.

38.1 Typographic Conventions

Throughout this documentation the authors use conventions to denote certain concepts, information, or actions. The following list gives examples of how to format these items in the documentation.

Menu Selections **Firewall > Rules**

GUI Item Labels/Names **Destination**

Buttons  **Add**

Prompt for input **Proceed?**

Input from the user (text) `Rule Description`

Input from the user (selection) *WAN*

File Names `/boot/loader.conf`

Names of commands or programs `gzip`

Commands Typed at a shell prompt

```
# ls -l
```

Long shell command-line examples may be split using the backslash (\) for shell line continuation.

Command Output

```
-rw-r--r--  2 root  wheel   887 Apr 12 09:49 .cshrc
```

Special Notes

Note: Consider this . . .

Warnings

> **Warning:** Watch out!

Tips

> **Tip:** The best practice is ...

References

See also:

For more information ...

The list above also servers as an example of a "definition list" used to define sets of terms or options and their meanings.

38.2 Documentation Quality Guidelines

38.2.1 Overview

This document aims to give an idea of what types of information should be in articles to keep them as useful as possible for other users of the documentation.

For information about style and formatting, see *Style Guide*.

38.2.2 What to include in articles

- Concise, detailed information - the more accurate the information, the more useful it is.
- Good spelling and grammar
- Good formatting (*Style Guide*)
- Proper categories to classify the article. For example, place the new article in the correct area of the documentation site (e.g. articles about OpenVPN go under `source/vpn/openvpn/`).

38.2.3 What not to include in articles

- Poor grammar
- Inconsistent or poor backing data
- Disparaging comments
- Ambiguity - try to make articles clear so there is no confusion
- Avoid creating a new directory or category for a single article if possible. If more will be added, then it is OK.
- Redundant information already covered in other documents. Especially in recipes.
- Do not start a recipe with installing pfSense software or other basic tasks. Make assumptions and link to other articles for topics already covered elsewhere. There is no need to reinvent the wheel and have multiple overlapping copies of the same information. Keep recipes as concise as possible!

38.2.4 Where to see why an article was flagged

Articles flagged for cleanup should have corresponding comments on the source of the article at GitHub, or issues on GitHub discussing what's wrong with them.

38.3 Style Guide

To make this documentation easier for users, we prefer to have the style of articles be consistent and clear. The following guidelines are strongly suggested. Text found to not be following these Language Style/Grammar guidelines may be edited and corrected at any time.

See *Documentation Quality Guidelines* for information about how entries in the documentation should be written and what they should contain.

38.3.1 Trademarks

The first use of **pfSense®** must have the ® symbol on each and every page.

38.3.2 Capitalization

Capitalize terms correctly! Especially **pfSense**! No other capitalization of "pfSense" may be used except in a URL which is acceptable as lowercase (e.g. https://www.pfsense.org). If a sentence begins with "pfSense" the first letter must remain lowercase.

If any other usage of "pfSense" or a misspelling of same is found ("PFsense", "PFSense", "pfSence", etc), fix it immediately.

Other special notes for capitalization:

- WebGUI, IPsec, OpenVPN, Internet, Ethernet, VPN, DNS, PPPoE, IPv4, IPv6, NPt, strongSwan, squidGuard, pfsync, pftop, JavaScript.

38.3.3 What to Avoid

Avoid addressing the user directly ("you", "your", etc.)** Rewrite sentences to avoid this usage when found. Exceptions may be made for quoted/cited text or other unavoidable circumstances.

Avoid references to the writer ("I", "we") Except when making specific recommendations, which is OK to avoid using passive voice. "We recommend" is better than "It is recommended".

Avoid the use of words such as "should", "could", "might" Words that do not commit to a specific action/result are undesirable. For example "This should happen" or "That might appear". Some instances are expected/required when making recommendations, but reword where feasible.

Avoid the use of Weasel words See Weasel Words for reference.

Avoid redundant phrases This especially includes acronym references that duplicate words: "WAN Network", "LAN Network", "DUID Identifier", "6RD Rapid Deployment". Remove the redundant word(s) and/or use alternate phrasing ("WAN Subnet" or "Network on the WAN interface" rather than "WAN Network")

Avoid unnecessary abbreviations and shortening of words This creates ambiguity, for example:

- Avoid using "IP" or "IPs" to refer to IP addresses. Use the full form "IP address" instead to remove ambiguity.

- Avoid using "config" when "config.xml" or "configuration" is more clear.
- Avoid using "ovpn" to mean "OpenVPN" except in cases when the OS-level interfaces are being referenced (`ovpncX` is OK, "Use OVPN instead" is not.)

Avoid using "here" for links Do not make links for "here", "click here", or similar phrasing. They provide no context for the link, cause redundancy in phrasing, and cause problems for users that require accessibility functions such as screen readers. See recommendations from W3C Tips and uxmovement.

Avoid awkward possessive references For example: "firewall's", "pfSense's".

Avoid gender-specific pronouns Example: "his", "hers"

Avoid leaving out necessary hyphens Example: "howto" should be "how-to"

Avoid "Britishisms" or other non-en_US style spellings Use "Flavor", not "Flavour". Use "Specialize", not "Specialise"

Avoid confusing bandwidth specifications

- Avoid confusing bits and bytes. Use Big **B** for bytes, little **b** for bits.
- Avoid ambiguity when specifying bandwidth measurements. Bandwidth should always have *time* component, such as *per second*. "5 Mbit/s" is much clearer than "5 Mbps", "5 Mb", or "5MB".

38.3.4 Crafting Instructions

When forming lists of instructions, start each item with an action word when possible: Click x, Select x, Enter x, Navigate to x.

When instructing a user to reach a specific page or place in the GUI, we prefer to reference this action as "Navigate to".

38.3.5 Example Text

When offering examples, keep the following in mind:

- Domain names should use `example.com` or another reserved name from RFC 2606.
- IP address examples should be taken from subnets reserved for documentation in RFC 6890: 192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24 or the traditional RFC 1918 networks 192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8 if the documentation subnets are not sufficient.
  - In some cases where additional unique examples are needed, use the benchmarking subnet 198.18.0.0/15.

38.3.6 Referring to items involving pfSense software

Refer to a firewall running pfSense as a "firewall" or "node". Avoid other similar terms ("router", "system", "box", etc.) for clarity and consistency.

38.3.7 High Availability / CARP References

- Refer to the cluster as a "High Availability Cluster" or "HA Cluster" and **not** as a "CARP Cluster".
- Use the term "node" as in "cluster node" for referencing an individual unit.
- Use the term "primary" for the primary node, never "master" as this can be confused with the CARP VIP status.
- Use the term "secondary" for the secondary node, never "backup" or "slave" as this can be confused with the CARP VIP status.
- Use the terms "Sync interface" or "Interconnect interface" when referring to the dedicated interface between HA Cluster nodes. **Never refer to that interface as a "CARP interface"**.

38.4 Formatting Guide

The pfSense® documentation is built using Sphinx/reStructuredText. The formatting is similar in some ways to Markdown, but has significant differences. To get a feel for the formatting, look at the source of this documentation and read through this document.

Tip: Test out how different markup is rendered using the Online reStructuredText editor. Additional information can be found at A primer on reStructuredText and reST/Sphinx cheat sheet.

38.4.1 Filenames

When adding new pages or images use all lowercase letters and hyphens instead of spaces for the filename. This is commonly referred to as a slug, or a slugified version of the text. For example, this file is named `formatting-guide.rst`.

Tip: Use an online slug generator.

38.4.2 Text

In general, try to keep text in logical paragraphs wrapped at 80 characters. This ensures the source is easy for everyone to read no matter where it is being edited. For long pages with several sections that may only be relevant to some users, split the page into several smaller documents.

Basic Inline Formatting

Add basic inline formatting to the text as follows:

- one asterisk: `*text*` for *emphasis* (italics),
- two asterisks: `**text**` for **strong emphasis** (boldface), and
- backquotes: ``` ``text`` ``` for `code samples`.
- Two colons at the end of a line (or on a blank line) to start a code block, prefix each line inside the block with two spaces:

```
code
```

These can be applied to text in various ways within the documentation:

- Menu references use bold text and ">" with spaces in between to separate menus from menu items: **System > General**
  - Navigation that also refers to a tab name should be formatted like so: **System > Advanced**, **Miscellaneous** tab.
- GUI text references and option names use bold text: **Description**
- Text to be entered or replaced by the user uses backquotes: Enter `192.168.1.1` for the **IPv4 Address**.
- Options selected by the user from a list or drop-down are italic: Select *WAN* for the the **Interface**.
- File names and paths use backquotes: `/root`
- Commands names inline with other text use backquotes, "The `sudo` command . . . ".
- Shell commands being demonstrated or directed use code blocks. Lead the line with two spaces then "#" to simulate a command prompt:

```
# ls -l /root
```

- Program output also uses code blocks, blank lines in output can either be blank or preceded by two spaces:

```
# someprogram
Output:

Foo
```

38.4.3 Headings

Headings consist of text and a line of characters underneath ("underline") the same length as the text. The specific characters must be consistent to denote sections of the same depth. Parts and chapters also use a similar row of characters above the text ("overline").

- `####` with overline, for parts
- `****` with overline, for chapters
- `====`, for sections
- `----`, for subsections
- `^^^^`, for subsubsections
- `""""`, for paragraphs

Note: The headings are also how the "On This Page" section is generated. When possible, it is a good idea to use headings that create an outline of the content, making it easy for the reader to scan.

38.4.4 Lists

Unordered Lists

Place an asterisk at the start of a paragraph and indent two spaces for any lines that wrap.

```
* This is a bulleted list.
* It has two items, the second
  item uses two lines.
```

Which renders as:

- This is a bulleted list.
- It has two items, the second item uses two lines.

Ordered lists

The same goes for numbered lists; they can will be auto-numbered using `#.`:

```
#. This is a numbered list.
#. It has two items too.
```

Which renders as:

1. This is a numbered list.
2. It has two items too.

Nested lists

Nested lists are possible, but be aware that they must be separated from the parent list items by blank lines:

```
* this is
* a list

  * with a nested list
  * and some subitems

* and here the parent list continues
```

Which renders as:

- this is
- a list
  - with a nested list
  - and some subitems
- and here the parent list continues

Definition lists

Definition lists are created as follows:

```
term (up to a line of text)
   Definition of the term, which must be indented

   and can even consist of multiple paragraphs

next term
   Description.
```

Note: The term itself cannot have more than one line of text.

Which renders as:

term (up to a line of text) Definition of the term, which must be indented

and can even consist of multiple paragraphs

next term Description.

Field Lists

Field lists are perfect for lists of options:

```
:Option Name: What it does.
:Option 2: Another option. This is a long description that wraps
  to the next line, with two spaces indentation.
:Third Option: Something else.
```

Which renders as:

Option Name What it does.

Option 2 Another option. This is a long description that wraps to the next line, with two spaces indentation.

Third Option Something else.

38.4.5 Links

External Link

Separate the link and the target definition, like this:

```
This is a paragraph that contains `a link`_.
```

```
.. _a link: http://example.com/
```

and place the target definition at the bottom of the page in alphabetical order.

Note: If the link text will contain a colon, escape it in both the link text and the definition, for example:

```
See `Link\: Stuff`_.
```

```
.. _Link\: Stuff: http://example.com/stuff
```

Cross Reference to Section of Document

To make a cross reference to another document, first create a label immediately before the section title:

```
.. _label-some-section:

Some Section
------------
```

And then in the other document, reference it using `:ref:` and the given label:

```
See :ref:`label-some-section` for more information
```

Cross Reference to Entire Document

If a cross-reference will instead reference an entire document rather than a specific section, use the `:doc:` method instead.

For example, to reference this entire document, `/references/style-guide.rst`, use the following text, omitting the file extension:

```
:doc:`/references/style-guide`
```

38.4.6 Images

Images

Place images in the `source/_static` directory in the same folder structure as the page that the image is going to be posted on. For example, an image going on `source/references/fomatting-guide.rst` would go in `source/_static/references/image.png`.

```
.. image:: /_static/filename.png
   :align: center
   :alt: Alternative text that describes the image
   :target: /_static/filename.png
```

Note: `:target:` is optional and only necessary if it is a large image.

Figures

Place figures in the `source/_static` directory in the same folder structure as the page that the image is going to be posted on. For example, an image going on `source/references/fomatting-guide.rst` would go in `source/_static/references/image.png`.

Figures are similar to images, but need a unique label and a caption for proper in-text references.

```
.. _figure-my-stuff:
.. figure:: /_static/stuff.png
   :figclass: align-center
   :target: /_static/stuff.png

   This is the caption
```

Which can be referred to using the following:

```
An example is shown in Figure :ref:`figure-my-stuff`.
```

Note: The indention is important! The caption *must* be aligned properly with the other attributes!

Inline Images

For an inline image (no breaks above or below, aka inline with the text) a substitution must be used. Since inline images are typically inserted on many pages, the inline image file can be placed in the root of `source/_static`.

Many common icon substitutions are available in a common substitutions file usable as follows:

```
.. include:: substitutions.rst
<lots of other text>
To add a blah, click |image_icon_plus|.
```

To do this in a one-off fashion, use a substitution within the same file:

```
Click |image_icon_edit| to edit the entry
<rest of page>
.. |image_icon_edit| image:: _static/icon_e.png
```

38.4.7 Tables

Grid Tables

The grid must be "painted", they look like this example:

```
+------------------------+------------+----------+----------+
| Header row, column 1   | Header 2   | Header 3 | Header 4 |
| (header rows optional) |            |          |          |
+========================+============+==========+==========+
| body row 1, column 1   | column 2   | column 3 | column 4 |
+------------------------+------------+----------+----------+
| body row 2             | ...        | ...      |          |
+------------------------+------------+----------+----------+
```

Simple Tables

These are easier to write, but are limited: they must contain more than one row, and the first column cells cannot contain multiple lines. They look like this:

```
=====  =====  =======
A      B      A and B
=====  =====  =======
False  False  False
True   False  False
False  True   False
True   True   True
=====  =====  =======
```

38.4.8 Table of Contents

Every file has to a part of a `toctree` or **Table of Contents** tree, as this is how the side navigation is built.

Reference RST files by their filenames without their `.rst` extension. It is also possible to link to external resources if necessary, as shown with the YouTube link:

```
.. toctree::
   :maxdepth: 2

   filename1
   filename2
   sub-directory/index
   Example YouTube Video <https://youtu.be/Cwz7vWu_KO0>
```

Local Table of Contents

Sometimes it is useful to add the table of contents of the current page:

```
.. contents:: :depth: 2
```

38.4.9 Colored Boxes

Admonitions are text, distinguished in friendly boxes, that bring attention to important items. The most common example is a "Note" box:

```
.. note:: This is a note, it will be surrounded by a note box when it is built.
```

Which renders as:

Note: This is a note, it will be surrounded by a note box when it is built.

Admonitions are available for a wide variety of types, including:

- `note`
- `tip`
- `warning`

- attention
- caution
- danger
- error
- hint
- important
- seealso

38.4.10 Substitutions

reST supports "substitutions", which are pieces of text and/or markup referred to in the text by `|name|`. They are defined like footnotes with explicit markup blocks, like this:

```
.. |name| replace:: replacement *text*
```

or this:

```
.. |caution| image:: warning.png
              :alt: Warning!
```

To use substitutions for multiple documents, put them into a separate file and include it into all documents where they will be used, using the `include` directive. Give the include file a file name extension differing from that of other source files, such as `.rsti`, to avoid Sphinx finding it as a standalone document.

A common substitutions file is available and is already referenced in a number of existing documents. Check that file before adding more substitutions in other files. Substitutions which will be widely used in many documents should be placed there.

38.4.11 Literal (code) Blocks

Briefly described earlier, literal or "code" blocks allow for pre-formatted text, most commonly used for source code, shell commands, command output, and so on.

A code block can be started by ending a sentence with two colons, and then a blank line. These two colons may also be on a line by themselves:

```
::

  code code code
```

The lines inside the code block must be indented to the same level, usually two spaces.

Blank spaces may be used between lines of code, they do not need to contain spaces.

For more complex examples, syntax highlighting can be used for source code using the `code-block` directive:

```
.. code-block:: html
   :linenos:

   <b>some html</b>
```

Which renders as:

```
<b>some html</b>
```

38.5 Updating Documentation

The primary avenue to make Netgate aware of problems in the documentation is to open an Issue on Redmine. In addition to requesting corrections, this may also be used to request new content or to propose other changes.

When creating an issue, please be specific and reference documents by filename where appropriate.

38.6 Developer Style Guide

This page covers rules and styles to be used when submitting code for inclusion in pfSense® software.

38.6.1 Modularity and Organization

Historically pfSense, and before it M0n0wall, have placed all of the functionality required to obtain configuration data, display it, edit it, validate it, format it, and save it in a single PHP file (web page). Much use was made of global variables to allow functions access to the required data. These are both undesirable strategies.

The "everything in one file" approach leads to large, difficult to manage files. Functions declared in PHP web pages cannot be accessed from outside that page leading to duplication, bloat and maintenance headaches. Global variables are fragile and often mysterious, particularly when the variable is declared in another file. What does "global $p_interface_remote;" mean? What type is it, who/what sets it, where is it declared, is it multi-user safe, etc?

pfSense functionality, particularly in PHP web pages, should adhere to MVC methodologies. The PHP web page file should contain only the code to display and edit information. The code to retrieve it, validate it, update it and save it should be provided in discrete functions included ("required") from a separate file (preferably in /usr/local/pfSense/include/www).

All functions should be organized in a modular fashion with clearly defined inputs and return values documented in the comments. The use of global variables should be minimized. Globals such as $g, $config and $input_errors etc are unavoidable and so acceptable, but where possible values should be passed in either directly or by reference.

References:

https://en.wikipedia.org/wiki/Modular_programming https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

38.6.2 Developer Rules

- Never commit untested code.
- If a developer breaks the code, they fix it, even if their code breaks another subsystem. This is not glamorous but it is the right thing to do (or back the code out until a proper fix can be obtained).
- If a developer commits a kernel change that requires a userland configuration change of any type, then that developer must also make sure there is PHP code to control the userland change as needed (different sysctls, different means of configuring something, upgrade code etc.).
- Avoid changing the XML configuration structure whenever possible. Where that is infeasible, do not commit code that changes the XML configuration structure without adding configuration upgrade code at the same time.

- Never push a commit to any branch that knowingly causes a regression.
- Any major or high risk changes must first be done in a git clone and reviewed by another committer before merging into mainline. Using a *pull request* for this workflow is acceptable, and is how every change from those outside the development team is handled.
- Pre-commit authorization is not used in pfSense software except in the case of major changes. For external/community developers, *pull requests* serve this function.
- Mention relevant ticket numbers in commit messages so that Redmine can associate the changes. See *Referencing Tickets in Commit Messages*.
- When possible, compose and format commit messages similar to entries in the change logs. For example:
  - Brief first sentence that describes the commit
  - Start with an action word describing the nature of the change ("Adds…", "Changes…", "Improves…", "Corrects…")
  - Reference the ticket number in the first sentence (see previous bullet point)
  - Examples:
    - "Change X to Y which fixes #1234"
    - "Correct check for XYZ in some_page.php to prevent badthing. Fixes #2345"
    - "Add coolnewthing to some_page.php. Implements #3456"
  - Longer, more detailed explanations can be placed on the next line and later.
- Always use full paths when calling an executable (e.g. `/usr/bin/grep` NOT `grep`)

38.6.3 HTML Specific Rules

Note: Incorrect HTML code is treated as broken code. Breaking the code is not allowed. A C compiler for example would complain in most cases if a developer breaks the code syntactically. Web browsers may ignore invalid code, but this does not mean that the code is not broken. The broken code must be fixed by the person who committed the invalid code.

---

pfSense software uses the XHTML doctype in its webGUI code. The doctype enforces code against the following ruleset:

- Use lower case tag names and not a mix of uppercase and lowercase tag names
- Breaks must be closed (`<br />`)
- Image tags must be closed (`<img />`)
- An image tag *always* has an alt attribute, though it may contain no value
- Horizontal rule tags must be closed (`<hr />`)
- HTML form fields must be closed (`<input />`)
- Ampersands in a URL (e.g. within a href attribute) must be coded as a HTML entity (e.g. `&`)
- Special characters (e.g. umlauts) must be coded as HTML Entities:
- A `<table />` tag does not have a name attribute
- A `<div />` tag does not have a name attribute
- A `<ul />` tag does not have a name attribute

- A tag does not have a name attribute
- Checkbox checked attributes must be coded as checked="checked"
- HTML field disabled attributes must be coded as disabled="disabled"
- HTML field readonly attributes must be coded as readonly="readonly"
- Any HTML <input /> field has a type attribute (e.g. type="text")
- Opening , <b> tags must have a matching closing tag (e.g.)
- <table /> tags do not contain a <form /> tag
- The type attribute of a <form /> tag must contain a lower case value (e.g. type="post" or type="get")
- The language=JavaScript attribute for <script /> tags is deprecated. Use the type="text/javascript" attribute instead.
- Always use lowercase attribute names for calling JavaScript events (e.g. onclick="foobar();")
- The <embed /> tag is deprecated, use <object /> instead
- If a style attribute is assigned to an HTML element it must be enclosed by quotes, for example: element.style.borderTop = "2px solid #990000";

It is possible to syntax check code in Firefox with the HTML validator plugin, or use the W3C validator. The latter is supported by Opera even for RFC 1918 networks.

38.6.4 PHP Specific Rules

Rule #1: Any time a rule is broken, there must be proper justification and documentation.

General rules

- All php files must start with a header block in English
- Use descriptive variable names in English
- Use lowercase variable names ($my_very_long_var_name) or Camel Case names ($myVeryLongVarName)
- When referencing variables inline in double quoted strings, use braces around the variable names:

```
$foo = "bar{$bar}bar";
```

- Add comments in English, whenever necessary or helpful
- Use // or /* */ style syntax for single line comments, do not use #
- Use /* */ style syntax for multi-line comments
- Use elseif and not else if when given a choice. The else if variant only works with braced syntax and not colon syntax (e.g. if: ... elseif: ... endif;).
- For testing the same variable against multiple strings or values directly, use a switch statement rather than a long chain of if/elseif/elseif/elseif/[...]/else statements.
- Add TODO: comments, when there is something to be done
- Add FIXME: comments, when something is broken

- add `NOTE:` comments, when there is something important other people should know beyond a traditional comment, for example a warning about not changing code in certain ways.
- Try to code in a readable way:

```
$header = "<head>{$foo}</head>";
$message = "SOME{$bar}TEXT";
```

Is easier to read than:

```
$header="<head>".$foo."</head>";
$message = "SOME" . $bar . "TEXT";
```

- Try to simplify code for better readability:

```
if ($bool1)
   if ($bool2)
       if ($bool3)
           do_it();
whatever();
```

- Should be written as:

```
if ($bool1 && $bool2 && $bool3) {
   do_it();
}
whatever();
```

- Do not set unnecessary or single-use variables:

```
$is_set = isset($var);
if ($is_set) ...
```

- Loop variables are `$i`, `$j`, `$k`, ...
  - Do **NOT** use `$g` for a loop variable, as it conflicts with the global `$g` used by pfSense software
- All `switch` statements must have a `default`
- In classes, use `private`, `protected` and `public`, not `var` for attribute declaration
- Do not to use deprecated or obsolete syntax or functions
  - Keep an eye on future versions of PHP to avoid using functions that will be deprecated in the future as well
- If a PHP-internal function is an alias for another function, use the original (i.e. use `exit()` instead of `die()`)

Indent style

- Use K&R, BSD KNF variant style:

```
if ($x == $y) {
   something();
   ...
} else {
   somethingelse();
   ...
}
finalthing();
```

- When creating `if`, `for`, `foreach`, and other similar block style structures, even if there is only one statement inside, the use of braces is required.

  For example, good:

```
if ($foo) {
   something();
}
```

  Not good:

```
if($foo)
   something();
```

- If a conditional statement must span multiple lines, indent using four spaces to align with the start of the conditional above it:

```
if ($foo1 && $foo2 && $foo3 && $foo4 && $foo5 && $foo6 &&
    $foo7 && $foo8 && $foo9) {
something();
}
```

- Do not put be a space between a function name and its argument list:

```
isset($myvar);
```

  - Conditional/control statements such as `if`, `foreach`, and `switch` are exceptions to this. Those must have a space before the parenthesis.

- . . . but **do** separate function arguments with a single space:

```
do_something($foo, 27, false);
```

- Use tabs for indentation – NOT spaces or a mixture of both
- . . . but spaces are OK in the middle of a line and for long conditional alignment
- Use a tab stop of 8, rather than 4, in an editor.
- Ensure there is NO trailing whitespace at the end of a line, for example spaces or tabs when there is no more text afterward
- Ensure there is NO whitespace on empty lines. For example, a line must not contain only spaces or only tabs

Configuration Manipulation

- Boolean values which are false should be un-set:

```
$config['system']['enablesshd'] = "no";
```

  should be:

```
unset($config['system']['enablesshd']);
```

38.6.5 JavaScript Specific Rules

- pfSense software does not support outdated browsers, so do not take special measures to use code required by old/obsolete browsers or rendering engines
- pfSense software includes, among other JavaScript resources, Bootstrap and jQuery. While native JavaScript is best for simple tasks, if a developer can accomplish a goal easily using an included library, they can use it instead
- pfSense software does not currently utilize `transpiler` or similar utilities
- Take special care with user input or statements/variables that can be populated with user input to avoid creating a vulnerability vector such as XSS. User fields must be encoded or otherwise sanitized
  - For example, be extremely cautions of values inserted into JavaScript via PHP variables. `json_encode()` can help avoid a situation where a user-supplied string could include text such as quotes or semicolons that leads to execution of arbitrary JavaScript

38.6.6 Shell Script Specific Rules

- Use braces in **all** variable references for proper parameter expansion:

```
${SOMETHING}
```

38.6.7 Ports/Packages Specific Rules

When working with the pkg system and FreeBSD ports structure, adhere to the FreeBSD guidelines for code in these files.

Useful resources for working with pkg and ports include:

- The FreeBSD Porter's Handbook
- The FreeBSD Ports bsd.port.mk file
- Use portlint to check the syntax of the Makefile and other supporting files
  - Install portlint on a FreeBSD system and run the following command inside the root directory of the port:

```
portlint -CN
```

- Run a the following command to make sure the contents of pkg-plist are correct:

```
make -DNO_DEPENDS check-plist
```

Other Guidelines:

- A port version or revision must increase for the port to be rebuilt, otherwise changes will not propagate to the pkg servers to be picked up by clients
  - For very minor changes, add or increase the `PORTREVISION` line immediately beneath `PORTVERSION` in the `Makefile`, starting at 1, for example: A second revision would be `PORTREVISION=2`
  - For more significant changes, increase `PORTVERSION`
    - When increasing `PORTVERSION`, completely remove any `PORTREVISION` line, do not comment it out
  - Do not add or change `PORTEPOCH` except under direction of a committer

38.6.8 External Code

Code that has been imported from an external source does not need to be changed to fit these guidelines.

38.6.9 Editor Configuration

The pfSense project uses a similar coding style to FreeBSD, which has editor configurations for Emacs and Vim. The FreeBSD man page style(9) contains additional relevant material.

CHAPTER

THIRTYNINE

CONFIGURATION RECIPES

CHAPTER

FORTY

ADDITIONAL COMMERCIAL RESOURCES

- Netgate TAC
- Netgate Professional Services
- pfSense Training

INDEX

D

DNS, **1515**

H

HTTP, **1515**

I

ICMP, **1515**
IP, **1515**

L

LAN, **1515**

N

NAT, **1515**

S

SSH, **1515**

T

TCP, **1515**

U

UDP, **1515**

V

VM, **1515**
VPN, **1515**

W

WAN, **1515**