
The pfSense Book

Release

The pfSense Team

May 10, 2017

CONTENTS

1	Preface	1
1.1	Acknowledgements	1
1.2	Feedback	3
1.3	Typographic Conventions	3
1.4	Authors	4
2	Foreword	7
3	Introduction	9
3.1	What does pfSense stand for/mean?	9
3.2	Why FreeBSD?	9
3.3	Common Deployments	10
3.4	Interface Naming Terminology	11
3.5	Finding Information and Getting Help	12
3.6	Project Inception	13
4	Networking Concepts	15
4.1	Understanding Public and Private IP Addresses	15
4.2	IP Subnetting Concepts	16
4.3	IP Address, Subnet and Gateway Configuration	16
4.4	Understanding CIDR Subnet Mask Notation	17
4.5	CIDR Summarization	18
4.6	Broadcast Domains	19
4.7	IPv6	19
4.8	Brief introduction to OSI Model Layers	32
5	Hardware	33
5.1	Minimum Hardware Requirements	33
5.2	Hardware Selection	33
5.3	Hardware Sizing Guidance	34
5.4	Hardware Tuning and Troubleshooting	38
5.5	Hardware Compatibility	39
6	Installing and Upgrading	41
6.1	Download Installation Media	41
6.2	Prepare Installation Media	43
6.3	Connect to the Console	48
6.4	Perform the Installation	51
6.5	Assign Interfaces	55
6.6	Alternate Installation Techniques	57
6.7	Installation Troubleshooting	58

6.8	Upgrading an Existing Installation	60
6.9	Filesystem Tweaks	63
7	Configuration	65
7.1	Setup Wizard	65
7.2	Interface Configuration	72
7.3	Managing Lists in the GUI	74
7.4	Quickly Navigate the GUI with Shortcuts	74
7.5	General Configuration Options	75
7.6	Advanced Configuration Options	77
7.7	Console Menu Basics	98
7.8	Time Synchronization	104
7.9	Troubleshooting	106
7.10	pfSense XML Configuration File	109
7.11	What to do when locked out of the WebGUI	110
7.12	Connecting to the WebGUI	114
8	Interface Types and Configuration	117
8.1	Interface Groups	117
8.2	Wireless	119
8.3	VLANs	119
8.4	QinQs	119
8.5	Bridges	119
8.6	OpenVPN	119
8.7	PPPs	120
8.8	GRE (Generic Routing Encapsulation)	123
8.9	GIF (Generic tunnel InterFace)	123
8.10	LAGG (Link Aggregation)	124
8.11	Interface Configuration	126
8.12	IPv4 WAN Types	128
8.13	IPv6 WAN Types	130
8.14	Physical and Virtual Interfaces	132
9	User Management and Authentication	133
9.1	User Management	133
9.2	Authentication Servers	136
9.3	External Authentication Examples	139
9.4	Troubleshooting	141
9.5	Support Throughout pfSense	143
10	Certificate Management	145
10.1	Certificate Authority Management	145
10.2	Certificate Management	148
10.3	Certificate Revocation List Management	152
10.4	Basic Introduction to X.509 Public Key Infrastructure	154
11	Backup and Recovery	157
11.1	Making Backups in the WebGUI	157
11.2	Using the AutoConfigBackup Package	157
11.3	Alternate Remote Backup Techniques	160
11.4	Restoring from Backups	161
11.5	Backup Files and Directories with the Backup Package	164
11.6	Caveats and Gotchas	165
11.7	Backup Strategies	166

12 Firewall	167
12.1 Firewalling Fundamentals	167
12.2 Ingress Filtering	169
12.3 Egress Filtering	169
12.4 Introduction to the Firewall Rules screen	172
12.5 Aliases	176
12.6 Firewall Rule Best Practices	183
12.7 Rule Methodology	185
12.8 Configuring firewall rules	190
12.9 Floating Rules	196
12.10 Methods of Using Additional Public IP Addresses	198
12.11 Virtual IP Addresses	201
12.12 Time Based Rules	203
12.13 Viewing the Firewall Logs	205
12.14 How Do I Block access to a Web Site?	209
12.15 Troubleshooting Firewall Rules	210
13 Network Address Translation	213
13.1 Port Forwards	213
13.2 1:1 NAT	219
13.3 Ordering of NAT and Firewall Processing	222
13.4 NAT Reflection	224
13.5 Outbound NAT	227
13.6 Choosing a NAT Configuration	230
13.7 NAT and Protocol Compatibility	231
13.8 IPv6 Network Prefix Translation (NPT)	234
13.9 Troubleshooting	236
13.10 Default NAT Configuration	239
14 Routing	241
14.1 Gateways	241
14.2 Gateway Settings	242
14.3 Gateway Groups	245
14.4 Static Routes	245
14.5 Routing Public IP Addresses	249
14.6 Routing Protocols	252
14.7 Route Troubleshooting	254
15 Bridging	259
15.1 Creating a Bridge	259
15.2 Advanced Bridge Options	259
15.3 Bridging and Interfaces	262
15.4 Bridging and firewalling	264
15.5 Bridging Two Internal Networks	265
15.6 Bridging interoperability	266
15.7 Types of Bridges	267
15.8 Bridging and Layer 2 Loops	268
16 Virtual LANs (VLANs)	269
16.1 Terminology	269
16.2 VLANs and Security	270
16.3 pfSense VLAN Configuration	271
16.4 Switch VLAN Configuration	275
16.5 pfSense QinQ Configuration	285
16.6 Requirements	287

17 Multiple WAN Connections	289
17.1 Multi-WAN Terminology and Concepts	289
17.2 Policy Routing, Load Balancing and Failover Strategies	291
17.3 Multi-WAN Caveats and Considerations	292
17.4 Summary of Multi-WAN Requirements	294
17.5 Load Balancing and Failover with Gateway Groups	294
17.6 Interface and DNS Configuration	296
17.7 Multi-WAN and NAT	297
17.8 Policy Routing Configuration	298
17.9 Verifying Functionality	300
17.10 Troubleshooting	302
17.11 Multi-WAN on a Stick	303
17.12 Multi-WAN for IPv6	303
17.13 Multi-Link PPPoE (MLPPP)	305
17.14 Choosing Internet Connectivity	306
18 Virtual Private Networks	309
18.1 Choosing a VPN solution	309
18.2 VPNs and Firewall Rules	311
18.3 VPNs and IPv6	312
18.4 PPTP Warning	313
18.5 Common deployments	313
19 IPsec	317
19.1 IPsec and IPv6	317
19.2 Choosing configuration options	317
19.3 IPsec and firewall rules	326
19.4 Site-to-Site	326
19.5 Mobile IPsec	335
19.6 Testing IPsec Connectivity	370
19.7 IPsec Troubleshooting	371
19.8 Configuring Third Party IPsec Devices	379
19.9 IPsec Terminology	382
20 OpenVPN	385
20.1 OpenVPN and IPv6	385
20.2 OpenVPN Configuration Options	385
20.3 Using the OpenVPN Server Wizard for Remote Access	396
20.4 Configuring Users	402
20.5 OpenVPN Client Installation	403
20.6 Site-to-Site Example (Shared Key)	415
20.7 Site-to-Site Example Configuration (SSL/TLS)	418
20.8 Checking the Status of OpenVPN Clients and Servers	422
20.9 Permitting traffic to the OpenVPN server	423
20.10 Allowing traffic over OpenVPN Tunnels	424
20.11 OpenVPN clients and Internet Access	424
20.12 Assigning OpenVPN Interfaces	424
20.13 NAT with OpenVPN Connections	426
20.14 OpenVPN and Multi-WAN	427
20.15 OpenVPN and CARP	431
20.16 Bridged OpenVPN Connections	431
20.17 Custom configuration options	433
20.18 Sharing a Port with OpenVPN and a Web Server	434
20.19 Controlling Client Parameters via RADIUS	434

20.20	Troubleshooting OpenVPN	435
20.21	OpenVPN and Certificates	439
21	L2TP VPN	441
21.1	L2TP and Firewall Rules	441
21.2	L2TP and Multi-WAN	441
21.3	L2TP Server Configuration	441
21.4	L2TP with IPsec	443
21.5	L2TP Troubleshooting	446
21.6	L2TP Logs	447
21.7	L2TP Security Warning	448
22	Traffic Shaper	449
22.1	What the Traffic Shaper can do for a Network	449
22.2	Hardware Limitations	450
22.3	ALTQ Scheduler Types	450
22.4	Configuring the ALTQ Traffic Shaper With the Wizard	453
22.5	Monitoring the Queues	461
22.6	Advanced Customization	462
22.7	Limiters	466
22.8	Traffic Shaping and VPNs	469
22.9	Troubleshooting Shaper Issues	470
22.10	Traffic Shaping Types	472
22.11	Traffic Shaping Basics	472
23	Server Load Balancing	473
23.1	Server Load Balancing Configuration Options	473
23.2	Web Server Load Balancing Example Configuration	477
23.3	Troubleshooting Server Load Balancing	483
24	Wireless	485
24.1	Recommended Wireless Hardware	485
24.2	Working with Virtual Access Point Wireless Interfaces	488
24.3	Wireless WAN	489
24.4	Bridging and wireless	492
24.5	Using an External Access Point	493
24.6	pfSense as an Access Point	495
24.7	Additional protection for a wireless network	501
24.8	Configuring a Secure Wireless Hotspot	502
24.9	Troubleshooting Wireless Connections	504
25	Captive Portal	507
25.1	Captive Portal Zones	507
25.2	Common Captive Portal Scenarios	507
25.3	Zone Configuration Options	508
25.4	MAC Address Control	516
25.5	Allowed IP Address	518
25.6	Allowed Hostnames	518
25.7	Vouchers	519
25.8	File Manager	523
25.9	Viewing Authenticated Captive Portal Users	524
25.10	Troubleshooting Captive Portal	525
25.11	Limitations	526
26	High Availability	529

26.1	pfsync Overview	529
26.2	pfSense XML-RPC Config Sync Overview	530
26.3	Example Redundant Configuration	530
26.4	Multi-WAN with HA	536
26.5	Verifying Failover Functionality	538
26.6	Providing Redundancy Without NAT	540
26.7	Layer 2 Redundancy	543
26.8	High Availability with Bridging	544
26.9	Using IP Aliases to Reduce Heartbeat Traffic	544
26.10	Interface	545
26.11	High Availability Troubleshooting	545
26.12	CARP Overview	549
27	Services	553
27.1	IPv4 DHCP Server	553
27.2	IPv6 DHCP Server and Router Advertisements	560
27.3	DHCP & DHCPv6 Relay	562
27.4	DNS Resolver	563
27.5	DNS Forwarder	568
27.6	Dynamic DNS	571
27.7	SNMP	574
27.8	UPnP & NAT-PMP	576
27.9	NTPD	578
27.10	Wake on LAN	583
27.11	PPPoE Server	585
27.12	IGMP Proxy	586
28	System Monitoring	589
28.1	System Logs	589
28.2	Remote Logging with Syslog	593
28.3	Dashboard	595
28.4	Interface Status	600
28.5	Service Status	601
28.6	Monitoring Graphs	601
28.7	Firewall States	609
28.8	Traffic Graphs	614
28.9	System Activity (Top)	615
28.10	pfInfo	615
28.11	S.M.A.R.T. Hard Disk Status	616
28.12	SMTP and Growl Notifications	621
28.13	Viewing the Contents of Tables	621
28.14	Testing DNS	622
28.15	Testing a TCP Port	622
29	Packages	625
29.1	Installing Packages	625
29.2	Reinstalling and Updating Packages	626
29.3	Uninstalling Packages	627
29.4	Developing Packages	627
29.5	A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid	627
29.6	Introduction to Packages	631
30	Third Party Software and pfSense	633
30.1	RADIUS Authentication with Windows Server	633
30.2	Syslog Server on Windows with Kiwi Syslog	639

30.3	Using Software from FreeBSD's Ports System (Packages)	639
30.4	Configure BIND as an RFC 2136 Dynamic DNS Server	641
31	Packet Capturing	645
31.1	Selecting the Proper Interface	645
31.2	Limiting capture volume	645
31.3	Packet Captures from the WebGUI	646
31.4	Using tcpdump from the command line	647
31.5	Using Wireshark with pfSense	655
31.6	Additional References	658
31.7	Capture frame of reference	658
32	Menu Guide	661
32.1	System	661
32.2	Interfaces	661
32.3	Firewall	662
32.4	Services	662
32.5	VPN	663
32.6	Status	663
32.7	Diagnostics	664

PREFACE

Acknowledgements

This book, and the pfSense project itself, would not be possible without a great team of developers, contributors, corporate supporters, and a wonderful community. The project has received code contributions from more than 200 people. Hundreds have contributed financially with hardware and other needed resources. Thousands more have done their part supporting the project by helping others on the mailing list, forum, and other platforms. And even more have contributed by purchasing hardware, support, and services such as the [pfSense Gold Subscription](#). Our thanks go out to everyone who has done their part to make the project the great success it has become.

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:

- Co-Founder 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

Personal Acknowledgements

From Chris

I must give my wife thanks and considerable credit for the completion of this book and the success of the project in general. This book and the project have led to countless long days, nights, and months without a day's break, and her support has been crucial.

I would also like to thank the many companies who have purchased our support and reseller subscriptions allowing me to make the jump to working full time on the project in early 2009.

I must also thank Jim for jumping in on this book and providing considerable help in completing it. It's been two years in the making, and far more work than I had imagined. It may have been obsolete before it got finished if it weren't for his assistance over the past several months. Also, thanks to Jeremy Reed, our editor and publisher, for his assistance with the book.

Lastly, my thanks to everyone who has contributed to the pfSense project in any fashion, especially the developers who have given huge amounts of time to the project over the past nine years.

From Jim

I would like to thank my wife and son, who put up with me throughout my participation in the writing process for not only the first book, but continued revisions over the years. Without them, I would have gone crazy a long time ago.

I would also like to thank Rick Yaney of HPC Internet Services, for being supportive of pfSense, FreeBSD, and Open Source software in general.

The entire pfSense community is deserving of even more thanks as well, it is the best and most supportive group of Open Source software users and contributors I have ever encountered.

Reviewers

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

- Jon Bruce

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

Feedback

The publisher and authors encourage feedback for this book and the pfSense distribution. Please send suggestions, criticism and/or praise for The pfSense Book to book@pfsense.org.

For general feedback related to the pfSense project, please post to the forum or mailing list. Links to these resources can be found at <http://pfsense.org/support>.

Typographic Conventions

Throughout the book the authors use conventions to denote certain concepts, information, or actions. The following list gives examples of how these are formatted in the book.

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 is an example of a “definition list” used to define sets of terms or options and their meanings.

Welcome to *The pfSense Book*, written by pfSense project co-founder Chris Buechler and coreteam member Jim Pingle. This book 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 book is designed to be a friendly guide to common networking and security tasks along with a thorough reference for the capabilities of pfSense software. *The pfSense Book* covers 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 a [Menu Guide](#) at the end of this book with all standard menu choices available in the pfSense software WebGUI.

Authors

Chris Buechler

One of the founders of the pfSense project, Chris was also one of its most active developers. He has been in the IT industry for over a decade, working extensively with firewalls and FreeBSD for most of that time. He has provided security, network, and related services for organizations in the public and private sector, ranging in size from small organizations to large public sector organizations and Fortune 500 companies. He holds numerous industry certifications including the CISSP, SSCP, MCSE, and CCNA amongst others. His personal web page can be found at <http://chrisbuechler.com>.

Jim Pingle

Jim has been working with FreeBSD for nearly 20 years and professionally for the past 15 years. Now a full-time employee of [Netgate](#), he provides global pfSense software support for customers, as well as software development and documentation. As a former system administrator for a local ISP in Bedford, Indiana USA, HPC Internet Services, he worked with FreeBSD servers, various routing equipment and circuits, and, of course, pfSense-based firewalls both internally and for many customers. Jim graduated in 2002 with a Bachelor's degree in Information Systems from Indiana-Purdue Fort Wayne.

When away from the computer, Jim enjoys spending time with his family, reading, playing board games and video games, woodworking, taking pictures, and being a television addict. His personal web page can be found at <http://pingle.org>.

FOREWORD

My friends and co-workers know that I build firewalls. At least once a month, someone says “My company needs a firewall with X and Y, and the price quotes I’ve gotten are tens of thousands of dollars. Can you help us out?”

Anyone who builds firewalls knows this question could be more realistically phrased as “Could you please come over one evening and slap together some equipment for me, then let me randomly interrupt you for the next three to five years to have you install new features, debug problems, set up features I didn’t know enough to request, attend meetings to resolve problems that can’t possibly be firewall issues but someone thinks might be the firewall, and identify solutions for my innumerable unknown requirements? Oh, and be sure to test every possible use case before deploying anything.”

Refusing these requests makes me seem churlish. Accepting these requests ruins my cheerful demeanor. For a long time, I wouldn’t build firewalls for anyone except for my employer.

pfSense lets me be a nicer person without having to actually work at it.

With pfSense, I can deploy a firewall in just a few hours with most of that time running cables and explaining the difference between “inside” and “outside.” pfSense’s extensive documentation and user community offers me an easy way to answer any questions with simply “did you look that up?”. If pfSense doesn’t support a feature then chances are I can’t support it either. But pfSense supports everything I could ask for – with a friendly interface to boot. The wide userbase means that features are tested in many different environments and generally *just works* even when interacting with the CEO’s kids’ Windows ME PC connected to the Internet by Ethernet over ATM over carrier pigeon. Best of all, pfSense is built on much of the same software I’d use myself. I trust the underlying FreeBSD operating system to be secure, stable, and efficient.

Security updates? Just click a button and reboot.

You need new features? Just turn them on. pfSense handles clustering, traffic shaping, load balancing, integration with your existing equipment through RADIUS, IPsec, monitoring, dynamic DNS, and more.

Big-name industry suppliers charge outrageous fees to support what pfSense freely provides. If your employer insists on paying for support contracts, or if you just feel more secure knowing you can pick up the phone and scream for help, you can get pfSense support agreements very reasonably. If you don’t need a support contract, I happen to know that Chris, Jim, or anyone else with a pfSense commit bit will let grateful pfSense users buy them a beer or six.

Personally, I don’t build firewalls from scratch any more. When I need a firewall, I use pfSense.

–Michael W. Lucas

INTRODUCTION

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.

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.

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.

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 as seen in this network performance analysis: <https://github.com/gvnn3/netperf/blob/master/Documentation/netperf.pdf>.

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.

Alternative Operating System Support

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

Common Deployments

pfSense 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.

Perimeter Firewall

The most common deployment of pfSense 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](#).

LAN or WAN Router

pfSense 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 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.

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 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 [IPv4 DHCP Server](#) and [IPv6 DHCP Server and Router Advertisements](#)

Interface Naming Terminology

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

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.

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.

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.

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*.

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](#).

Finding Information and Getting Help

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

Finding Information

The search function on the book is the easiest way to find information on a specific topic. The most common features and deployments of pfSense are covered in this book. When reading the HTML version of the book, 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 pfsense.org websites. The best way to search the sites is a Google search appending `site:pfsense.org` to the query. This will search the website, forum, the Redmine site, open access documentation, etc. which are all official sources of information.

Getting Help

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

The pfSense project offers several other ways to get help, including a [forum](#), [open access documentation](#), mailing lists and IRC (Internet Relay Chat, `##pfSense` on `irc.freenode.net`). Commercial support is also available on the [pfSense Portal](#). More information can be found on the pfSense website at [Obtaining Support](#). Many of these links are reachable from the GUI web configurator under the Help menu in pfSense.

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. In fact, the majority of users have never used FreeBSD outside of pfSense. In addition to being a powerful, flexible firewalling and routing platform, pfSense 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](#).

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.

NETWORKING CONCEPTS

Understanding Public and Private IP Addresses

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. 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 4.1: 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 4.2: 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.

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 book 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.

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 4.3: 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

Throughout the book, we use 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.

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](#).

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, 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 \(VLANs\)](#).

Understanding CIDR Subnet Mask Notation

pfSense uses CIDR (Classless Inter-Domain Routing) notation rather than the common subnet mask 255.x.x.x when configuring addresses and networks. Refer to the following [Table CIDR Subnet Table](#): to find the CIDR equivalent of a decimal subnet mask.

Table 4.4: 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
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 2.3.3-RELEASE supports the use of /31 networks for interfaces and Virtual IP addresses

So 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.

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 4.5: 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.

Subnet Mask Converter

Enter the dotted decimal Subnet Mask

or Enter the number of bits in the subnetmask

Fig. 4.1: 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.

Network/Node Calculator

Enter the Subnet Mask: 255 255 252 0

Enter the TCP/IP Address: 10 70 65 0 Calculate

Network: 10 70 64 0

Node/Host: 0 0 1 0

Broadcast Address: 10 70 67 255 Explain

Fig. 4.2: Network/Node Calculator

Finding a matching CIDR network

IPv4 Ranges in the format of x.x.x.x-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.

Broadcast Domains

A broadcast domain is the portion of a network sharing the same layer 2 segment. 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.

A single broadcast domain *can* contain more than one IPv4 or IPv6 subnet, however, that is generally not considered good network design. IP subnets should be segregated into separate 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 but care must be taken to avoid switch loops in this scenario. There are also some 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. More information on broadcast domains and how to combine them can be found in [Bridging](#).

IPv6

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.

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.

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](#) or [SixXS](#).

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.

IPv6 WAN Types

Details can be found in [IPv6 WAN 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.

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.

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 4.6: 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
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

Continued on next page

Table 4.6 – continued from previous page

Prefix	Subnet Example	Total IP Addresses	# of /64 nets
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.

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 4.7: IPv6 Special Networks and Addresses

Network	Purpose
2001:db8::/32	Documentation prefix, used for examples, like those find in this book.
::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

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**.

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.

Address Allocation

Client addresses can be allocated by static addressing through SLAAC (*Router Advertisements (Or: “Where is the DHCPv6 gateway option”)*), DHCP6 (IPv6 DHCP Server and 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*).

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 pfSense 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.

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.

Connecting with 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](#) or [SixXS](#). 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.

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**.

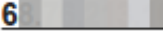
 Tunnel ID: 298327	Delete Tunnel
 Creation Date:	Jul 17, 2015
 Description:	
IPv6 Tunnel Endpoints	
 Server IPv4 Address:	184.105.253.14
 Server IPv6 Address:	2001:470:1f10:c4f::1/64
 Client IPv4 Address:	6  3
 Client IPv6 Address:	2001:470:1f10:c4f::2/64
Routed IPv6 Prefixes	
 Routed /64:	2001:470:1f11:c4f::/64
 Routed /48:	2001:470:c614::/48 [X]
Available DNS Resolvers	
 Anycasted IPv6 Caching Nameserver:	2001:470:20::2
Anycasted IPv4 Caching Nameserver:	74.82.42.42

Fig. 4.3: 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.

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**

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.

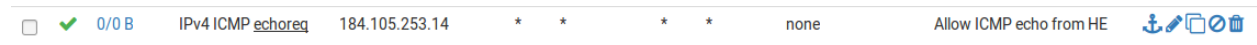


Fig. 4.4: Example ICMP Rule

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 > (assign)** 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 > (assign)**, **Interface Assignments** tab.
- Select the newly created GIF under **Available Network Ports**.
- Click  **Add** to add it as a new interface.

GIF Configuration	
Parent Interface	WAN This interface serves as the local address to be used for the GIF tunnel.
GIF Remote Address	184.105.253.14 Peer address where encapsulated gif packets will be sent.
GIF tunnel local address	2001:470:1f10:c4f::2 Local gif tunnel endpoint.
GIF tunnel remote address	2001:470:1f10:c4f::1 Remote GIF address endpoint.
GIF tunnel subnet	64 The subnet is used for determining the network that is tunnelled.
Route Caching	☐ Specify if route caching can be enabled. (Be careful with these settings on dynamic networks.)
ECN friendly behavior	☐ ECN friendly behavior violates RFC2893. This should be used in mutual agreement with the peer.
Description	HE Tunnel Broker A description may be entered here for administrative reference (not parsed).

Fig. 4.5: Example GIF Tunnel

Configure the New OPT Interface

The new interface is now accessible under **Interfaces > OPT x** , where x depends on the number assigned to the interface.

- Navigate to the new interface configuration page. (**Interfaces > OPT x**)
- 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**.

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**.

General Configuration	
Enable	☒ Enable interface
Description	WANv6 Enter a description (name) for the interface here.
IPv4 Configuration Type	None
IPv6 Configuration Type	None

Fig. 4.6: Example Tunnel Interface

Edit Gateway	
Disabled	☐ Disable this gateway Set this option to disable this gateway without removing it from the list.
Interface	WANV6 Choose which interface this gateway applies to.
Address Family	IPv6 Choose the Internet Protocol this gateway uses.
Name	WANV6_TUNNELV6 Gateway name
Gateway	dynamic Gateway IP address
Default Gateway	☒ This will select the above gateway as the default gateway.

Fig. 4.7: 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*.

WANV6_TUNNELV6	2001:470:1f10:c4f::1	2001:470:1f10:c4f::1	94.468ms	7.145ms	0.0%	Online	Interface WANV6_TUNNELV6 Gateway

Fig. 4.8: Example Tunnel Gateway Status

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.

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.

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 DHCP Server and 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.

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.

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*.

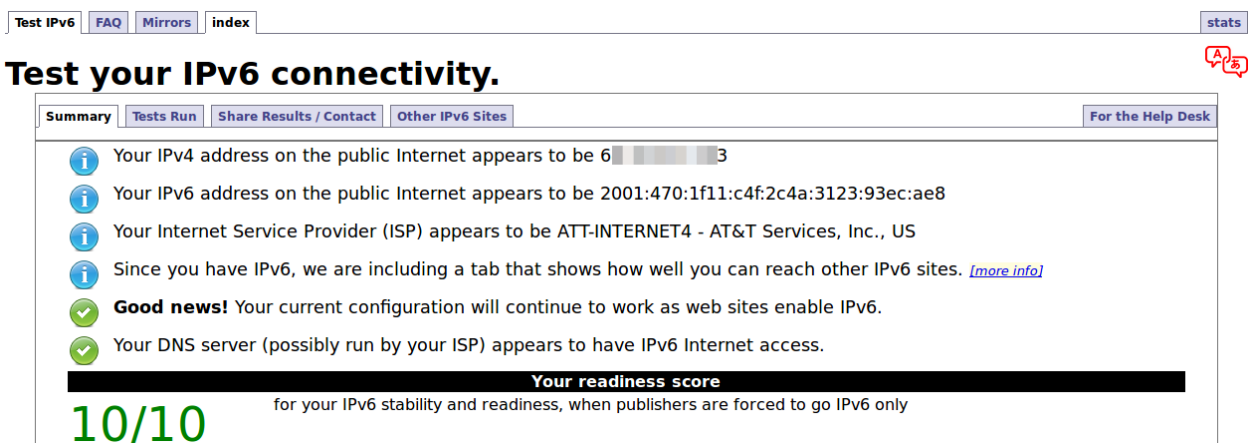


Fig. 4.9: IPv6 Test Results

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.

Controlling IPv6 Preference for traffic from the firewall itself

By default, pfSense 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.

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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the July 2015 Hangout on IPv6 Basics

This book 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.

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 book 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.

HARDWARE

Minimum Hardware Requirements

The minimum hardware requirements for pfSense 2.3.3-RELEASE are:

- CPU 600 MHz or faster
- RAM 512 MB or more
- 4 GB or larger disk drive (SSD, HDD, etc)
- One or more compatible network interface cards
- Bootable USB drive or CD/DVD-ROM for initial installation

Note: The minimum requirements are not suitable for all environments; see [Hardware Sizing Guidance](#) for details.

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.

Preventing hardware headaches

Use Official pfSense Hardware

We highly recommend using official hardware from the [pfSense Store](#). Our hardware has been developed to assure the community that specific hardware platforms have been thoroughly tested and validated.

Search for the experiences of others

The experiences of others are a valuable source knowledge which can be found by researching pfSense and hardware compatibility online, especially on the [pfSense Forum](#). Reports of failure should not necessarily be 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:pfsense.org` will search the pfSense.org website for relevant user experiences. Searching for the make, model, and pfSense will find user experiences on other websites or the mailing list archives. Repeating the same search with FreeBSD instead of pfSense may also turn up useful experiences.

Naming Conventions

Throughout this book we refer 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.

The SG-1000 firewall is based on the armv6 architecture, but since the SG-1000 is currently the only ARM device compatible with pfSense, we will refer to SG-1000 by name rather than using a generic ARM type or architecture name.

Hardware Sizing Guidance

When sizing hardware for pfSense, required throughput and necessary features are the primary factors that govern hardware selection.

Throughput Considerations

The minimum requirements are enough if less than 100 Mbps of unencrypted throughput is required. For higher throughput requirements, follow these guidelines based on extensive testing and deployment experiences.

The hardware referenced in this section is available from the [pfSense Store](#). The general specifications of the hardware can be found in Table *pfSense Hardware*.

Table 5.1: pfSense Hardware

Model	CPU	Clock	Cores	RAM	NIC Speed
SG-1000	TI ARM Cortex-A8 AM3352	600 MHz	1	512 MB	2x1G
SG-2220	Intel Rangeley Atom C2338	1.7 GHz	2	2 GB	2x1G
SG-2440	Intel Rangeley Atom C2358	1.7 GHz	2	4 GB	2x1G
SG-4860	Intel Rangeley Atom C2558	2.4 GHz	4	8 GB	6x1G
SG-8860	Intel Rangeley Atom C2758	2.4 GHz	8	8 GB	6x1G
XG-2758	Intel Rangeley Atom C2758	2.4 GHz	8	16 GB	2x10G, 4x1G
C2758	Intel Rangeley Atom C2758	2.4 GHz	8	8 GB	4x1G +2x10G
XG-1540	Intel Xeon-DE D-1541	2.0 GHz	8	16-32 GB	2x10G, 2x1G

The Table *Maximum Throughput by CPU - 1400 Byte Packets* compares throughput between several hardware models. The table contains data for Packets Per Second (PPS) and throughput in Mbit/s for TCP and UDP. A packet size of 1400 bytes was used for this test.

Warning: The numbers below are sample results of ongoing testing and do not accurately reflect the full potential throughput of each device.

Table 5.2: Maximum Throughput by CPU - 1400 Byte Packets

Model	UDP - 1400 Byte Packets		TCP - 1400 Byte Packets	
	PPS	Mbit/s	PPS	Mbit/s
SG-1000				
SG-2220				
SG-2440				
SG-4860				
SG-8860				
XG-2758				
C2758				
XG-1540				

As shown by the above results, at this relatively large packet size most models can reach the maximum speed of their fastest available network card type, accounting for overhead.

For contrast, look at the same statistics but this time using 64 byte packets, which contain the smallest possible payload. These are shown in Table *Maximum Throughput by CPU - 64 Byte Packets*

Warning: The numbers below are sample results of ongoing testing and do not accurately reflect the full potential throughput of each device.

Table 5.3: Maximum Throughput by CPU - 64 Byte Packets

Model	UDP - 64 Byte Packets		TCP - 64 Byte Packets	
	PPS	Mbit/s	PPS	Mbit/s
SG-1000				
SG-2220				
SG-2440				
SG-4860				
SG-8860				
XG-2758				
C2758				
XG-1540				

With smaller packet sizes, the packets per second rate is generally higher, but typically with lower overall throughput. This is because the hardware has to work much harder to move the larger volume of smaller packets.

In real networks the traffic flow will likely fall between these sizes, but it completely depends on the environment and the type of traffic involved. 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 5.4: 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.

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 5.5: 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. 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.

Hardware crypto accelerators greatly increase maximum VPN throughput and largely eliminate the performance difference between accelerated ciphers. Table [VPN Throughput by hardware model, all values are Mbit/s](#) illustrates the maximum throughput for various hardware available from the pfSense store when using IPsec and OpenVPN.

For IPsec, 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.

Warning: The numbers below are sample results of ongoing testing and do not accurately reflect the full potential throughput of each device.

Table 5.6: VPN Throughput by hardware model, all values are Mbit/s

Model	OpenVPN / AES-128+SHA1		IPsec / IKEv2+AES-GCM	
	TCP, 64B	TCP, 1400B	TCP, 64B	TCP, 1400B
SG-1000				
SG-2220				
SG-2440				
SG-4860				
SG-8860				
XG-2758				
C2758				
XG-1540				

Where high VPN throughput is a requirement for a firewall, hardware crypto acceleration is of utmost importance to ensure not only fast transmission speeds but also less 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 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.

Suricata is multi-threaded and can potentially take advantage of NETMAP for inline IPS if the hardware offers support.

Squid

Squid is a caching HTTP proxy server available as a pfSense package. Disk I/O performance is an important consideration for Squid users since it determines cache performance. In contrast, disk speed is largely irrelevant for most pfSense firewalls since the only significant disk activity is boot time and upgrade time; it has no relevance to network throughput or other normal operation.

Even for Squid any hard drive will suffice in small environments. For 200+ user deployments, a high-quality SSD from a Tier-1 OEM is a must-have. A low-quality SSD may not be able to endure the numerous writes involved with maintaining cached data.

The size of the squid cache also factors into the amount of RAM required for the firewall. Squid consumes approximately 14MB of RAM per 1GB of cache on amd64. At that rate, a 100GB cache would require 1.4GB of RAM for cache management alone, not counting other RAM needs for squid.

Hardware Tuning and Troubleshooting

The underlying operating system beneath pfSense can be fine-tuned in many ways. A few of these tunables are available in pfSense under **Advanced Options** (See [System Tunables Tab](#)). Others are outlined in the FreeBSD main page [tuning\(7\)](#).

The default installation of pfSense software 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 [pfSense Store](#) is tuned further since we have detailed knowledge of the hardware, removing the need to rely on more general assumptions.

Common changes along these lines for other hardware can be found in the documentation wiki page for [Tuning and Troubleshooting Network Cards](#).

Note: Changes to `/boot/loader.conf.local` require a firewall reboot to take effect.

Mbuf Exhaustion

A common problem encountered by users of commodity hardware is mbuf exhaustion. 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. This is more common with NICs that use multiple queues or are otherwise optimized for performance over resource usage. Also, 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"
```

Additionally, cards 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 network card types 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

Another common issue is a NIC not properly supporting MSIX despite its claims. MSIX can be disabled by adding the following line to `/boot/loader.conf.local`:

```
hw.pci.enable_msix=0
```

The pfSense software distribution is compatible with most hardware supported by FreeBSD.

pfSense version 2.4 and later is compatible with 64-bit (amd64, x86-64) architecture hardware and the SG-1000 ARM-based firewall.

pfSense version 2.3.x and before was compatible with 32-bit (i386, x86) and 64-bit (amd64, x86-64) architecture hardware.

Alternate hardware architectures such as ARM (other than SG-1000), PowerPC, MIPS, SPARC, etc. are not supported at this time.

Hardware Compatibility

The best way to ensure that hardware is compatible with pfSense software is to buy hardware from the [pfSense Store](#) that has been tested and known to work well with pfSense. The hardware in the store is tested with each release of pfSense software and is tuned for performance.

For home-built solutions, the FreeBSD Hardware Notes is the best resource for determining hardware compatibility. pfSense version 2.3.3-RELEASE is based on 10.3-RELEASE, so compatible hardware is found in the [hardware notes on the FreeBSD website](#). Another good resource is the [Hardware section of the FreeBSD FAQ](#).

Network Adapters

A wide variety of wired Ethernet cards (NICs) are supported by FreeBSD, and are thus compatible with pfSense firewalls. However, not all network adapters are created equal. The hardware can vary greatly in quality from one manufacturer to another.

We recommend Intel PRO/1000 1Gb and PRO/10GbE 10Gb NICs because they have solid driver support in FreeBSD and they perform very well. Most hardware sold in the [pfSense Store](#) contains Intel NICs.

Of the various other PCIe/PCI cards supported by FreeBSD, some work fine. Others may have problems such as VLANs not working properly, not being able to set speed or duplex, or poor performance. In some cases, FreeBSD may support a particular NIC but, with specific implementations of the chipset, driver support may be poor. When in doubt, search the [pfSense 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

We do not recommend using USB network adapters of any make/model due to their unreliability and poor performance.

Wireless Adapters

Supported wireless adapters and recommendations are covered in [Wireless](#).

INSTALLING AND UPGRADING

Hardware from the [pfSense 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 [pfSense 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 [Installation Troubleshooting](#).

This chapter also covers upgrading pfSense software installations ([Upgrading an Existing Installation](#)) which keeps them up-to-date with the latest security, bug fixes, and new features.

Download Installation Media

Customers who have purchased firewalls from the [pfSense Store](#) can download tuned factory installation images from their account on the [pfSense Portal](#). The [Netgate Documentation](#) site contains specific instructions for each model, so check that site first before downloading images based on the information in this chapter.

For hardware from the pfSense store that no longer has support, or for other hardware, continue reading.

- Navigate to www.pfsense.org in a web browser on a client PC.
- Click **Downloads**.
- Select a **File Type** of *Install*.
- Select an **Architecture**:

AMD64 (64-bit) For 64-bit x86-64 Intel or AMD hardware.

Netgate ADI For most of the SG-series firewalls from the [pfSense Store](#), specifically, models which contain a USB console port on COM2.

- Select a **Platform** for a 64-bit install:

USB Memstick Installer A disk image which can be written to a USB memory stick (memstick) and booted on the target hardware for installation.

CD 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.

Serial Installs using a serial console on COM1 of the target hardware. This option requires a physical console port.

- Select a **Mirror** that is close to the client PC geographically.
- Click  **Download**.
- Copy the SHA-256 sum displayed by the page to verify the download later.

Note: 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**.

Tip: For older versions of pfSense software, look in the [Software Archive](#)

The file names for pfSense software version 2.3.3-RELEASE are:

USB Memstick Installer (Netgate ADI) `pfSense-CE-memstick-ADI-2.3.3-RELEASE-<arch>.img.gz`

USB Memstick Installer (VGA) `pfSense-CE-memstick-2.3.3-RELEASE-<arch>.img.gz`

USB Memstick Installer (Serial) `pfSense-CE-memstick-serial-2.3.3-RELEASE-<arch>.img.gz`

ISO Image installer `pfSense-CE-2.3.3-RELEASE-<arch>.iso.gz`

See also:

At any point in the installation if something does not go as described, check [Installation Troubleshooting](#).

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 verification on Windows

Windows users can install [HashTab](#) or a similar program to view SHA-256 hashes for any given file. The generated SHA-256 hash can be compared with the SHA-256 sum from the download site or the contents of the `.sha256` file from the download server. The `.sha256` file is viewable in any plain text editor such as Notepad.

With HashTab installed, to check the hash of a file:

- Right click on the downloaded file.
- Click the **File Hashes** tab. HashTab will take a few moments to calculate the hash.

- Hover the mouse over the SHA-256 hash to view the full hash.
- Paste the SHA-256 sum from the download site or from the `.sha256` file into the **Hash Comparison** box to automatically check if the hash matches.
- Click **Cancel** to dismiss the file properties dialog without making changes.

If a SHA-256 hash is not displayed in HashTab:

- Click Settings
- Check the box for **SHA-256**
- Click OK

Hash verification on BSD and Linux

The `sha256` command comes standard on FreeBSD and many other UNIX and UNIX-like operating systems. A SHA-256 hash can be generated by running the following command from within the directory containing the downloaded file:

```
# sha256 pfSense-CE-memstick-ADI-2.3.3-RELEASE-amd64.img.gz
```

Compare the resulting hash with the SHA-256 sum displayed on the download site or the contents of the `.sha256` file downloaded from the pfSense website. GNU or Linux systems provide a `sha256sum` command that works similarly.

Hash verification on OS X

OS X also includes the `sha256` command, the same as FreeBSD, but there are also GUI applications available such as [QuickHash](#). [HashTab](#) is also available for OS X.

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.

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.

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 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 to the 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/dal`, 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-2.3.3-RELEASE-amd64.img of=/dev/sdb bs=4M
```

- FreeBSD:

```
sudo dd if=pfSense-CE-memstick-ADI-2.3.3-RELEASE-amd64.img of=/dev/dal bs=4m
```

- Mac OSX:

```
sudo dd if=pfSense-CE-memstick-ADI-2.3.3-RELEASE-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 CD/DVD

To use an ISO image file with an optical disk drive, the ISO image must be burned to a CD or 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 CD containing the single ISO file. Burning procedures vary by OS and available software.

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 CD burning software package for Windows includes the ability to burn ISO images. Refer to the documentation for the CD 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 CD or 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.

Burning with Nero To burn an ISO image with Nero:

- Open Windows Explorer and locate the decompressed ISO image file
- Right click the ISO image file
- Click **Open With**
- Select **Nero**
- Follow the prompts in Nero to write the disc

The first time Nero is used, it may be necessary to select it from the **Choose Default Program** list. This process may work with other commercial CD burning software as well.

Burning with ISO Recorder If the client PC is using Windows XP, 2003, or Vista, the freely available [ISO Recorder](#) tool can write ISO images to disc.

- Download and install the appropriate version of ISO Recorder
- Browse to the folder on the drive containing the decompressed ISO image file
- Right click the ISO image file
- Click **Copy image to CD**

Other Free Burning Software Other free options for Windows users include [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 CD burning application that can handle ISO images.

If a CD 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-2.3.3-RELEASE-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 CDs in FreeBSD, see the CD 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 CD image and **not** as a data file.

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.

Connecting to a Serial Console

The instructions in this section cover general serial console topics. Some devices, such as firewalls from the [pfSense Store](#), require slightly different methods to connect to the serial console. For devices from the [pfSense 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

For most of the firewalls purchased from the [pfSense 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.

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. PC Engines ALIX defaults to 38400/8/N/1 and Soekris hardware defaults to 19200/8/N/1. 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**

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
```

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 [Installation Troubleshooting](#).

The following items are requirements to run the installer:

- [Download Installation Media](#)
- [Prepare Installation Media](#)
- [Connect to the Console](#)

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.

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

First, the installer prompts to launch the **Install** process or a **Rescue Shell**. To continue installing, press `Enter` while **Install** is selected. The **Rescue Shell** option 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.

The **Keymap Selection** screen is next. 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, find it in the list and select it instead. After making a selection, return to the top of the list and either choose **Test** or **Continue**.

Partition / Filesystem Selection

The **Partitioning** step selects the filesystem for the firewall's target disk. In pfSense 2.3.x and before, the only option was **UFS**. The new **ZFS** filesystem type is more reliable and has more features than the older UFS format, 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 process varies slightly depending on the selected filesystem type, so follow the section below that matches the filesystem type used by this firewall.

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 [Installation Troubleshooting](#) for help.

UFS

- Select **Auto (UFS)**
- 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.
- Select **Entire Disk**
- Select **Yes** to confirm that the installer can overwrite the entire disk
- Select the partition scheme to use for the disk:
 - GPT** The GUID partition table layout. Used by most modern 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.
 - Others** The other choices are not relevant to hardware that is capable of running pfSense software.
- Select **Finish** to accept the automatic partition layout chosen by the installer.

Note: The partition sizes and such can be customized here, but we do not recommend taking that step. For nearly all installations, the default sizes are correct and optimal.

- Select **Commit** to write the partition layout to the target disk.
- Skip ahead to *Continue with the Install*.

ZFS

- Select **Auto (ZFS)**
- Select **Pool Type / Disks**
 - Select 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.

stripe A single disk, or multiple disks added together to make one larger disk. **For firewalls with a single target disk, this is the correct choice.** (RAID 0)

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 the disks to use with the selected **Virtual Device Type**. Use the up and down arrow keys to highlight a disk and `Space` to select disks. **Select a disk even if there is only one in the list.** For mirrors or RAID types, select enough disks to fulfill the requirements for the chosen type.
- Select OK with the left and right arrow keys.
- 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. **Try this method first.**

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.

GPT + Lenovo Fix (BIOS) GPT with a Lenovo-specific boot fix.

- Change the default swap size (optional) by selecting **Swap Size** and then entering a new value. Typically the optimal size is 2x the available RAM in the firewall, but with smaller disks that may be too much.
 - Leave the other options on the screen at their default values.
 - Move the selection back to **Install** and ensure **Select** is highlighted 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.
 - Skip ahead 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 pfSense software installation is complete!

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 [pfSense 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).

```
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:b4    (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]? n

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 or a):
```

Fig. 6.1: Interface Assignment Screen

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.

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.

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

Alternate Installation Techniques

This section describes alternate methods of installation that may be easier for certain rare hardware requirements.

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.

Full 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.

Installation Troubleshooting

The vast majority of the time, installations will finish with no problems. If issues pop up, the following sections describe the most common problems and the steps to resolve them.

Boot from Install Media Fails

Due to the wide array of hardware combinations in use, it is not uncommon for a CD or memstick to boot incorrectly (or not at all). Given the unpredictable nature of commodity hardware support, using hardware from the [pfSense 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.

There are more troubleshooting techniques listed on the pfSense documentation Wiki under [Boot Troubleshooting](#).

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.

As in the previous section, there are more troubleshooting techniques listed in the online documentation under [Boot Troubleshooting](#).

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.

Hardware Troubleshooting

the following suggestions will help resolve general hardware issues.

Bootting from USB

If the boot stops with a `mountroot>` prompt while booting off the live CD, 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.

Upgrading an Existing Installation

pfSense software can be reliably upgraded from an older release to a current release.

By keeping a firewall running pfSense software updated with a current supported release, it will never be obsolete. We periodically release new versions that contain new features, updates, bug fixes, and various other changes. In most cases, updating a pfSense installation is easy. If the firewall is updating to a new release that is a only a point release (e.g 2.3.2 to 2.3.3), the update is typically minor and unlikely to cause problems.

The most common problem encountered during upgrades is 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 2.1.5 to 2.3.3 must be handled with care, and ideally tested on identical hardware in a test environment prior to use in production.

We post upgrade notes along with releases to help guide through any potential upgrade pitfalls. These notes vary from release to release, the most current version can be found on the [Upgrade Guide](#).

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. Customers with a [pfSense Gold Subscription](#) should consider using the **Auto Config Backup** package. Customers using the Auto Config Backup package can make a manual backup with a note identifying the change, which is encrypted and stored on our servers.

Another good practice is to have install media handy for the release currently being run, and 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](#).

Upgrading

There are several methods available for updating a normal installation of pfSense software. Either the WebGUI or the console can be used.

Upgrading using the WebGUI

The **Automatic Update** feature contacts a pfsense.org 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**.

Click  **Confirm** on **System > Update** to start the update if one is available.

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.

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. Once connected to the console of the firewall, start the upgrade process by choosing menu option 13.

Alternately, from a shell prompt running as root, manually execute the following command:

```
# pfSense-upgrade
```

Older Versions

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. However, for the time being, update files in that format are still provided by the project in order to bring older firewalls up to date.

On these older versions, an auto update will still function. After running the automatic update there may be newer releases available, so once the firewall is running a version of pfSense 2.4 or later, run another update if the firewall detects it is necessary.

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.

Update Settings

Branch / Tracking Snapshots

By default, the update check only looks for officially released versions of pfSense software, but this method can also be used to track development snapshots. The update location can be changed by visiting **System > Update, Update Settings** tab and selecting a different **Branch** in the **Firmware Branch** section.

Stable versions are the best option, as they see the most testing and are reasonably safe and trouble-free. However, as with any upgrade, visit the pfSense website and read the update notes for that release, and check the [Upgrade Guide](#).

Choose *Development Snapshots* to switch a firewall over to tracking development snapshot builds. These generally are snapshots for the next minor maintenance branch release.

In some cases, a *Next Major Version* option will be in the list. This option makes the firewall track 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. We do not generally recommend running these in production.

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.

Filesystem Tweaks

The default settings for the filesystem are best for most environments, however there are occasions that call for slight changes to improve stability, performance, or longevity of the filesystem.

Enabling TRIM Support

The installer for the factory version of pfSense sets TRIM automatically. Both factory and the CE version of pfSense 2.4 support TRIM natively when using ZFS.

While it is possible to manually enable TRIM, support is unpredictable in hardware so we do not provide instructions on how to enable the feature.

Triggering a Filesystem Check

pfSense will run a filesystem check (`fsck`) at boot when it detects an unclean 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.

In these cases:

- Connect to the console
- Choose the menu option to reboot from the console menu (5)
- Enter `F` (uppercase “f”) to force a filesystem check during the boot sequence even if the drive is considered clean

That option is not present on all firewalls as it is not compatible with certain BIOS implementations. If that option is not present:

- Reboot the firewall into single user mode by choosing option 2 from the boot menu
- Press `Enter` when prompted for a shell
- Enter `fsck -y /`
- Repeat the command at least 3 times, or until no errors are found, even if the filesystem is reported clean

CONFIGURATION

Setup Wizard

The first time a user logs into pfSense, the firewall presents the Setup Wizard automatically. The first page of the wizard is shown in Figure *Setup Wizard Starting Screen*.

Click  **Next** to start the configuration process using the wizard.

Tip: Using the setup wizard is optional. Click the pfSense logo at the top left of the page to exit the wizard at any time.

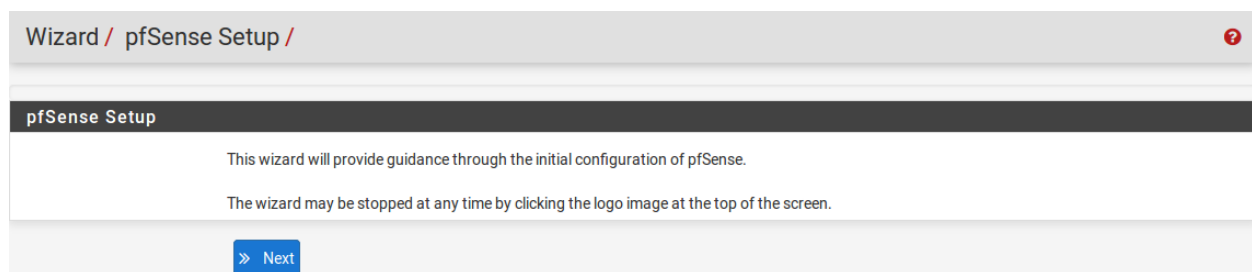


Fig. 7.1: Setup Wizard Starting Screen

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** 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>.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 of this firewall.

Primary/Secondary DNS Server The IP address of the Primary DNS Server and Secondary DNS Server may be filled in, if required and if they are known.

These DNS servers may be left blank if the DNS Resolver will remain active using its default settings. The default pfSense 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.

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.

Fig. 7.2: General Information Screen

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 `0.pfsense.pool.ntp.org`. This value will pick a random server from a pool of known-good NTP hosts.

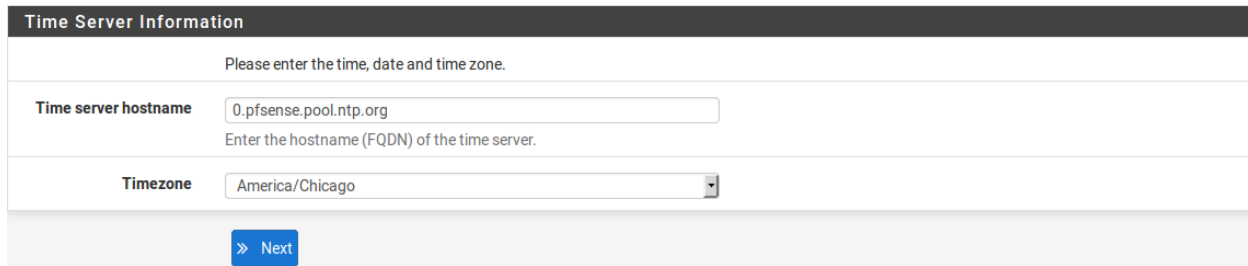
To utilize multiple time servers, add them in the same box, separating each server by a space. For example, to use three NTP servers from the pool, enter:

```
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.

Click  **Next** to continue.



Time Server Information

Please enter the time, date and time zone.

Time server hostname
Enter the hostname (FQDN) of the time server.

Timezone

 **Next**

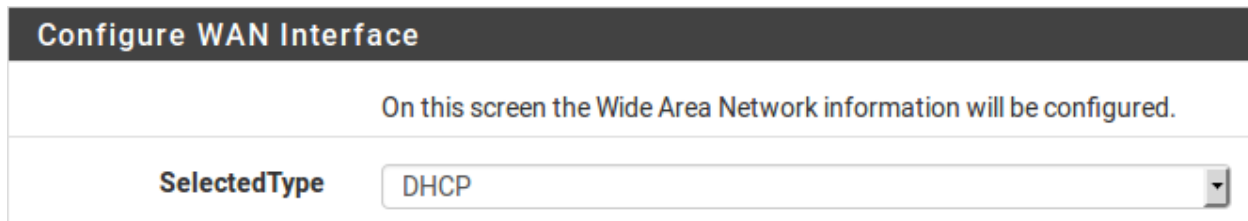
Fig. 7.3: NTP and Time Zone Setup Screen

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.



Configure WAN Interface

On this screen the Wide Area Network information will be configured.

SelectedType

Fig. 7.4: 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 hardware’s original 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.

General configuration	
MAC Address	 This field can be used to modify ('spoof') the MAC address of the WAN interface (may be required with some cable connections). Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank.
MTU	 Set the MTU of the WAN interface. If this field is left blank, an MTU of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.
MSS	 If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect. If this field is left blank, an MSS of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed. This should match the above MTU value in most all cases.

Fig. 7.5: General WAN Configuration

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.

Static IP Configuration	
IP Address	
Subnet Mask	24
Upstream Gateway	

Fig. 7.6: Static IP Settings

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.

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.

DHCP client configuration

DHCP Hostname

The value in this field is sent as the DHCP client identifier and hostname when requesting a DHCP lease. Some ISPs may require this (for client identification).

Fig. 7.7: DHCP Hostname Setting

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 Causes pfSense to leave 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 pfSense will let 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.

PPPoE configuration

PPPoE Username

PPPoE Password

Show PPPoE password

☐ Reveal password characters

PPPoE Service name

Hint: this field can usually be left empty

PPPoE Dial on demand

☐ Enable Dial-On-Demand mode

This option causes the interface to operate in dial-on-demand mode, allowing a virtual full time connection. The interface is configured, but the actual connection of the link is delayed until qualifying outgoing traffic is detected.

PPPoE Idle timeout

If no qualifying outgoing packets are transmitted for the specified number of seconds, the connection is brought down. An idle timeout of zero disables this feature.

Fig. 7.8: 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**.

PPTP configuration	
PPTP Username	
PPTP Password	
Show PPTP password	☐ Reveal password characters
PPTP Local IP Address	
pptplocalsubnet	32
PPTP Remote IP Address	
PPTP Dial on demand	☐ Enable Dial-On-Demand mode This option causes the interface to operate in dial-on-demand mode, allowing a virtual full time connection. The interface is configured, but the actual connection of the link is delayed until qualifying outgoing traffic is detected.
PPTP Idle timeout	 If no qualifying outgoing packets are transmitted for the specified number of seconds, the connection is brought down. An idle timeout of zero disables this feature.

Fig. 7.9: 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.

RFC1918 Networks	
Block RFC1918 Private Networks	☒ Block private networks from entering via WAN When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.
Block bogon networks	
Block bogon networks	☒ Block non-Internet routed networks from entering via WAN When set, this option blocks traffic from IP addresses that are reserved (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and obviously should not appear as the source address in any packets received.

Fig. 7.10: Built-in Ingress Filtering Options

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. 7.11: LAN Configuration

Click  **Next** to continue.

Set admin password

Next, change the administrative password for the WebGUI 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 WebGUI or SSH is exposed to the Internet, intentionally or accidentally, the firewall could easily be compromised if it still uses the default password.

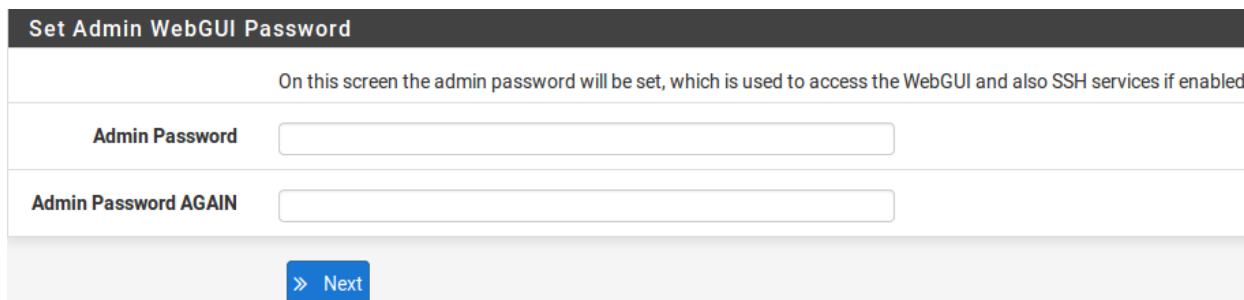


Fig. 7.12: Change Administrative Password

Completing the Setup Wizard

That completes the setup wizard configuration. Click **Reload** (Figure [Reload pfSense WebGUI](#)) and the WebGUI will apply the settings from the wizard and reload services changed by the wizard.

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`.

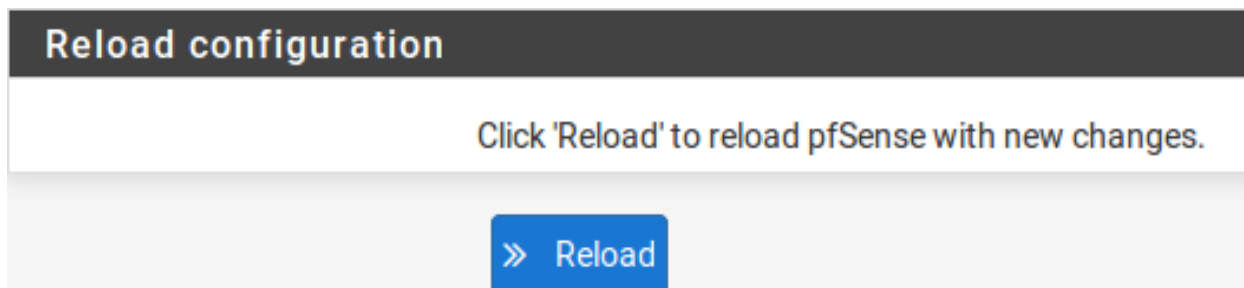


Fig. 7.13: Reload pfSense WebGUI

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 WebGUI.

Interface Configuration

Basic aspects of interface configuration 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](#).

Assign interfaces

Additional interfaces added after the initial setup may be assigned roles by visiting **Interfaces > (assign)**. 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 > (assign)**
- 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 > (assign)**
- 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*.

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**. Nearly all of the options found under **Interfaces > WAN** are identical to those mentioned in the WAN portion of the **Setup Wizard**.

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 is 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 cause pfSense to treat that 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.

Managing Lists in the GUI

The pfSense WebGUI 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 book for specifics about icons found in other parts of the firewall. For example, to find the meaning of icons used only in certificate management, look in [Certificate Management](#)

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.

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 a service's status page, logs, and configuration. The shortcuts for a given topic are present on every page related to that topic.

For example, in Figure [Shortcuts Example](#), the shortcuts have the following effects:

-  **Start Service** If the service is stopped, this icon starts the service.
-  **Restart Service** If the service is running, this icon restarts the service.

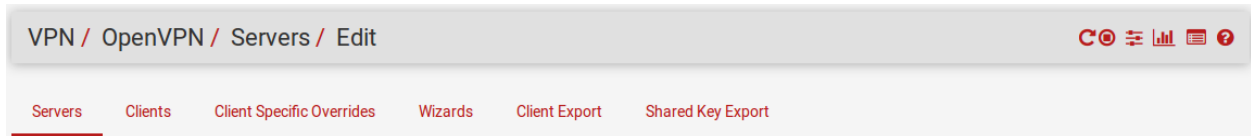


Fig. 7.14: Shortcuts Example

-  **Stop Service** If the service is running, this icon stops the service.
-  **Related Settings** When this icon appears, it navigates to the settings page for this section.
-  **Status Page Link** A link to the status page for this section, if one exists.
-  **Log Page Link** If this section has a related log page, this icon links there.
-  **Help Link** Loads 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.

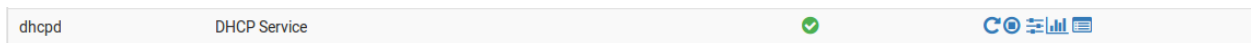


Fig. 7.15: Shortcuts on Service Status

General Configuration Options

System > General Setup contains options which set basic configuration items for pfSense and the GUI. 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 Enter 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`.

DNS Server Settings

Options in this section control how the firewall resolves hostnames using DNS.

DNS Server 1-4 Address The IP addresses of the DNS Servers may be filled in, if required and if they are known.

These DNS servers may be left blank if the DNS Resolver will remain active using its default settings. The default pfSense 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.

See also:

For more information on configuring the DNS Resolver, see [DNS Resolver](#)

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.

DNS Server 1-4 Gateway In addition to their IP addresses, this page provides a way to set the gateway used to reach each DNS server. This is especially 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 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 configured manually, uncheck this option.

Disable DNS Forwarder By default, pfSense 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.

Localization

Options in this section control the firewall's clock display and language.

Timezone 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 may require a reboot to fully activate in all areas of the firewall.

Time Servers 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 Servers** value at the default 0.pfsense.pool.ntp.org. This value will pick a random server from a pool of known-good NTP hosts.

To utilize multiple time 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.

Language The pfSense GUI has been translated into two other languages in addition to the default *English* language. The alternate languages are *Portuguese (Brazil)* and *Turkish*.

webConfigurator

Options in this section control various aspects of the GUI's behavior.

Theme Changing 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 is scrolled, the navigation remains at the top of the *page*, so when scrolling down 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 firewall's **Hostname** or **Fully Qualified Domain Name** will be included 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, more columns can be added to make better use of horizontal screen space. The maximum number of columns is 4.

Associated Panels Show/Hide A few areas of the pfSense GUI contain collapsible panels with settings. These panels take up extra screen space, so they are hidden by default. For firewall administrators that use the panels frequently, this can be slow and inefficient, so the options in this group allow the panels to be shown by default instead of hidden.

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**.

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.

Dashboard Update Period Controls the interval at which the dashboard data is updated. Many of the widgets update dynamically using AJAX. With many widgets loaded, a fast update interval can cause a high load on the firewall, depending on the hardware in use. Allowing longer time between updates would reduce the overall load.

Advanced Configuration Options

System > Advanced contains numerous options of an advanced nature. Few of these options require adjustment for basic routing/NAT deployments, these options can help customize the firewall configuration in beneficial ways for more complex environments.

Some of these options are covered in more detail in other sections of the book where their discussion is more topical or relevant, but they are all mentioned here with a brief description.

Admin Access Tab

The options found on the **Admin Access** tab govern the various methods for administering the firewall, including via the web interface, SSH, serial, and physical console.

webConfigurator (WebGUI)

Protocol

The WebGUI Protocol may be set to either **HTTP** or **HTTPS**. The best practice is to use HTTPS so that traffic to and from the WebGUI is encrypted.

SSL Certificate

If HTTPS is chosen, a certificate must also be chosen from the **SSL Certificate** drop-down list. The default certificate is an automatically-generated self-signed certificate. That is not an ideal situation, but is better than no encryption at all.

Tip: To use an externally signed SSL certificate and key, import them using the **Certificate Manager**, then select the certificate here.

The main downside to using a custom self-generated 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 have been cracking down on how they are handled. Firefox, for example, gives a warning screen and forces the user to import the certificate and allow a permanent exception. Internet Explorer will show a warning screen with a link to continue, as does Chrome. Opera will show a warning dialog.

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

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. Moving the GUI to another port will free up the standard web ports for use with port forwards or other services such as a HAproxy. By default the WebGUI 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.

Max Processes

If multiple administrators view the GUI at the same time and pages are taking too long to load, or failing to load, then increase the Max Processes value. By default it is set to 2, so the firewall runs two web server worker processes.

WebGUI Redirect

By default, for ease of access and compatibility, 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 HTTPS on port 443. This redirect can be disabled by checking **Disable webConfigurator redirect rule**. Disabling the redirect also allows another daemon to bind to port 80.

WebGUI Login Autocomplete

For convenience, the login form allows autocomplete so browsers can save the login credentials. In high-security environments, such as those that must adhere to specific security compliance standards, this behavior is not acceptable. It can be disabled by checking **Disable webConfigurator login autocomplete**. This only controls autocomplete on the login form.

Warning: Few browsers respect this option. Many of them will still offer to save passwords even when the form specifies that it should not be allowed. This behavior must be controlled or changed using browser options.

WebGUI login messages

Successful logins result in a message being printed to the console, and on some hardware these console messages cause a “beep” to be heard from the device. To stop this log message (and the resulting beep), check **Disable logging of webConfigurator successful logins**.

Anti-lockout

Access to the WebGUI port and SSH port on the LAN interface is permitted by default regardless of user-defined filter rules, due to the anti-lockout rule. When two or more interfaces are present, the anti-lockout rule is active on the LAN interface; If only one interface is configured, the anti-lockout rule will be active on that interface instead.

Checking **Disable webConfigurator anti-lockout Rule** removes the automatic lockout prevention rule. With that rule disabled, it is possible to control which LAN IP addresses may access the WebGUI using firewall rules.

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 system console also resets the anti-lockout rule. If administrative access is locked out after enabling this, 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

The firewall blocks private IP address responses from configured DNS servers by default, to prevent DNS rebinding attacks. Check this box to disable DNS rebinding protection if it interferes with webConfigurator access or name resolution.

See also:

More detail on DNS rebinding attacks may be found on [Wikipedia](#).

The most common case for disabling this would be when the firewall is set to use an internal DNS server which will return private (RFC1918) answers for hostnames. When accessing the firewall by IP address, these checks are not enforced because the attack is only relevant when using a hostname.

Tip: Instead of disabling all DNS rebinding protections, it 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

The GUI checks the referring URL when it is accessed to prevent 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 desirable convenience behavior, such as having a page that links to various firewall devices. To disable this behavior, check **Disable HTTP_REFERER enforcement check**.

Alternate Hostnames

To keep **DNS Rebind Checks** and **HTTP_REFERER Enforcement** active, but control their behavior slightly, fill in **Alternate Hostnames** in the box. By default the system will allow access to the hostname configured on the firewall and all IP addresses configured on the firewall. Adding hostnames in this field will allow those hostnames to be used 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, a message will be printed that indicating that access to the firewall may be compromised due to a Man-In-The-Middle (MITM) attack.

If such a forwarding was deliberately configured on the firewall or on a firewall ahead of this one, 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 in this case, only a warning, so there is no option to disable this behavior.

Browser Tab Text

By default, the firewall 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 can be enabled which allows 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 iTerm. To login to the admin account, either the `admin` username or `root` account may be used, and both accept the admin WebGUI password for login.

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 pfSense firewall are also possible by using a Secure Copy (SCP) client such as OpenSSH's 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 pfSense will change. Outdated clients may not be able to connect using the strong security keys and algorithms required by `sshd` on pfSense. 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.

Authentication Method

SSH can be configured to only allow key-based logins and not a password. Key-based logins are a much more secure practice, though it does take more preparation to configure.

To force key-based authentication, check **Disable Password login for Secure Shell**.

User keys for key-based login are added 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 their account.

SSH Port

Moving the SSH server to an alternate port provides a negligible security improvement, and frees up the port for other uses. To change the port, type the new port into the **SSH Port** box.

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 moving the SSH service to an alternate random port and forcing key-based authentication. Moving to an alternate port will prevent log noise from many, but not all, brute-force SSH login attempts and casual scans. It can still be found with a port scan, so switching to key-based authentication must always be done on every publicly accessible SSH server to eliminate the possibility of successful brute force attacks.

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.

Serial Communications

If pfSense 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 console is enabled on the first serial port. This console will receive the 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 115200bps 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 from pfSense. 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 console menu is always shown on the system 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 allows the console to be password protected. The same username and password may be used here that is used for the WebGUI. 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.

Firewall/NAT Tab

Firewall Advanced

IP Do-Not-Fragment compatibility

This option is a workaround for operating systems that 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 systems.

When this option is enabled, the firewall will not drop these malformed packets but instead clear the don't fragment 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.

Disable Firewall

When **Disable all packet filtering** is set, the pfSense firewall is turned into 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, pfSense 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 how many states are 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.

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 pfBlocker 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. pfSense 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: We recommend creating 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.

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: We always recommend using 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. Only TCP port forwards are supported.

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.

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.

Networking Tab

IPv6 Options

Allow IPv6

When the **Allow IPv6** option is unchecked, all IPv6 traffic will be blocked.

This option is checked by default on new configurations so that the firewall is capable of transmitting and receiving IPv6 traffic if the rules allow it to pass. This option controls a set of block rules that prevent IPv6 traffic from being handled by the firewall to allow compatibility with configurations imported from or upgraded from versions of pfSense older than 2.1.

Note: This option does not disable IPv6 functions or prevent it from being configured, it only controls traffic flow.

IPv6 over IPv4 Tunneling

The **Enable IPv4 NAT encapsulation of IPv6 packets** option enables IP protocol 41/RFC 2893 forwarding to an IPv4 address specified in the **IP address** 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 will cause 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.

Network Interfaces

Device Polling

Warning: Device polling has a detrimental effect on modern hardware and is unnecessary, decreasing performance and causing various problems. The option has been removed from pfSense as of version 2.4.

Device polling is a technique that lets the system periodically poll network devices for new data instead of relying on interrupts. This prevents the firewall WebGUI, SSH, etc. from being inaccessible due to interrupt floods when under extreme load, at the cost of higher latency. The need for polling has been nearly eliminated thanks to operating system advancements and more efficient methods of interrupt handling such as MSI/MSIX.

Note: With polling enabled, the system will appear to use 100% CPU. This is normal, as the polling thread is using CPU to look for packets. The polling thread is run at a lower priority so that if other programs need CPU time, it will give it up as needed. The downside is that this option makes the CPU graph less useful.

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.

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.

We recommend allowing 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.

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

Sticky Connections

When pfSense is directed to perform load balancing, successive connections will be redirected in a round-robin manner to a web server or gateway, balancing the load across all available servers or paths. When **Sticky Connections** is active this behavior is changed so that connections from the same source are sent to the same web server or through the same gateway, rather than being sent in a purely round-robin manner.

Sticky Connections affects both outbound load balancing (Multi-WAN) as well as server load balancing when enabled. This “sticky” association will exist as long as states are in the table for connections from a given source address. Once the states for that source expire, so will the sticky association. Further connections from that source host will be redirected to the next web server in the pool or the next available gateway in the group.

For outgoing traffic using a load balancing gateway group, the sticky association is between the user and a gateway. As long as the local address has states in the state table, all of its connections will flow out of a single gateway. This can help with protocols such as HTTPS and FTP, where the server may be strict about all connections coming from the same source, or where an additional inbound connection must be received 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.

For server load balancing, described further in [Server Load Balancing](#), sticky connections are desirable for applications which rely on the same server IP addresses being maintained throughout a given session for a user. Web applications on servers may not be intelligent enough to allow a user session to exist on multiple backend servers at the same time, so this allows a user to always reach the same server so long as they are browsing a site. This behavior may not be required depending on the content of the server.

Tip: For more control over how user connections are associated with servers in a load balancing scenario, consider using the HAProxy package instead of the built-in `relayd` load balancer. HAProxy supports several methods of ensuring users are properly directed to a backend server.

The **Source Tracking Timeout** for sticky connections 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.

Default Gateway Switching

Enable Default Gateway Switching allows additional non-default gateways to take over if the default gateway becomes unreachable. This behavior is disabled by default. With multiple WANs, switching the default gateway automatically will ensure that the firewall always has a default gateway so that traffic from the firewall itself can get out to the Internet for DNS, updates, packages, and add-on services such as squid.

Tip: When using the DNS Resolver in the default non-forwarding mode, default gateway switching is required for Multi-WAN to function properly. If default gateway switching cannot be used, then consider using forwarding mode instead.

There are cases where switching the default gateway is not desirable, however, such as when the firewall has additional gateways that are not connected to the Internet. In the future this option will be expanded so it can be controlled on a per-gateway basis.

Warning: This option is known to not work properly with a PPP-type WAN (PPPoE, L2TP, etc) as a default gateway.

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 less power considerably. 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. One optional module is selectable here: *AES-NI* (Advanced Encryption Standard, New Instructions). If *AES-NI CPU-based Acceleration* (`aesni`) is chosen, then its kernel module will be loaded when saved, and at bootup. 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. Some OpenSSL-based software like OpenVPN can perform differently with AES-NI unloaded since OpenSSL has built-in support for AES-NI. IPsec support will be greatly increased with AES-NI loaded provided that AES-GCM is used and properly configured.

These drivers hook into the `crypto(9)` framework in FreeBSD, so many aspects of the system will automatically use acceleration for supported ciphers.

There are other supported cryptographic devices, such as *hifn(4)* and *ubsec(4)*. In most cases, if a supported accelerator chip is detected, it will be shown in the **System Information** widget on the dashboard.

Thermal Sensors

pfSense 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.

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.

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

Clear States When a Gateway is Down

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*.

Note: 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 the firewall's disks. 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 hard 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.

/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.

/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.

Periodic RRD Backup The time, in hours, between periodic backups of RRD files. If the firewall is rebooted unexpectedly, the last backup is restored when the firewall boots back up. The lower the value, the less data that will be lost in such an event, but more frequent backups write more to the disk.

Periodic DHCP Leases Backup The time, in hours, between periodic backups of the DHCP lease databases. If the firewall is rebooted unexpectedly, the last backup is restored when the firewall boots back up. The lower the value, the less data that will be lost in such an event, but more frequent backups write more to the disk.

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.

The system logs are held in `/var` but they are not backed up like the RRD and DHCP databases. The logs will reset fresh on each reboot. For persistent logs, utilize remote syslog to send the logs to another device on the network.

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.

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. In almost all cases, we recommend leaving 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 book, but for those interested in digging a little deeper, The [sysctl](#) manual page from FreeBSD contains detailed instructions and information.

Notifications

pfSense notifies the administrator of important events and errors by displaying an alert in the menu bar, indicated by the  icon. pfSense can also send these notifications remotely via E-mail using SMTP or via Growl.

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, SMTP notifications will not be sent. 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 notifications will be sent.

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 have a valid SSL certificate and accept SSL/TLS connections.

From e-mail address The e-mail address that will be used in the `From:` header, which specifies the source of the message. 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** 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 pfSense 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.

Growl

Growl provides an unobtrusive method of delivering desktop notifications. These notifications pop up on a desktop and then hide or fade away. Growl is available in the App store for [Mac OSX](#), and it is available on [Windows](#) and [FreeBSD/Linux](#) as well.

Disable Growl Notifications When checked, the firewall will not send Growl notifications.

Registration Name The service name the firewall will use to register with the Growl server. By default this is `pfSense-Growl`. Consider this as the *type* of notification as seen by the Growl server.

Notification Name The name of the system that produces notifications. The default value of `pfSense growl alert` may be sufficient, or customize it with the firewall hostname or any other value to make it distinct.

IP Address The IP address to which the firewall will send Growl notifications.

Password The password required by the Growl server.

Click  **Save** to store the settings before proceeding.

Click  **Test Growl Settings** to send a test notification via Growl using the previously saved settings. Save before attempting a test.

Console Menu Basics

Basic configuration and maintenance tasks can be performed from the 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:

```
*** Welcome to pfSense 2.4.0-RELEASE (amd64) on pfsense ***

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
```

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.

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.

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.

Reset to factory defaults

This menu choice restores the system configuration to factory defaults. It will also attempt to remove any installed packages.

Note: This action will not make any other changes to the filesystem. 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.

This action is also available in WebGUI at **Diagnostics > Factory Defaults**

Reboot system

This menu choice will cleanly shutdown the pfSense firewall and restart the operating system.

A few advanced options may also be displayed on this page, depending on hardware support:

Reboot normally Performs a normal reboot in the traditional way.

Reroot This option does not perform a full reboot, but a “reroot” style boot. 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 This will restart 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 menu option is not available on SG-1000.

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 This reboots the firewall and forces a filesystem check using `fsck`, run five times. This operation can typically correct issues with the filesystem on the firewall. This menu option is not available on SG-1000.

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.

This action is also available in WebGUI at **Diagnostics > Reboot**

Halt system

This menu choice cleanly shuts down the firewall and either halts or powers off, depending on hardware support.

Warning: We strongly discourage cutting power from a running system. Halting before removing power is always the safest choice.

This action is also available in WebGUI at **Diagnostics > Halt System**

Ping host

This menu option runs a script which attempts to contact a host to confirm if it is reachable through a connected network. The script prompts the user for an IP address, and then it 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.

Shell

This menu choice starts a command line shell. A shell is very useful and very powerful, but also has the potential to be very dangerous.

Note: The majority of pfSense 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 a pfSense system 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, we do not recommend or support using other shells.

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.

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 logs in real time with low detail, use this shell command:

```
clog -f /var/log/filter.log | filterparser.php
```

Restart webConfigurator

Restarting the webConfigurator will restart the system process that runs the WebGUI (`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.

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 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.

disablereferercheck

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 on the [pfSense Doc Wiki](#).

installpkg / listpkg / uninstallpkg

These scripts interface with the pfSense 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 the `relayd` load balancer 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.

restartdhcpd

This script stops and restarts the DHCP daemon.

restartipsec

This script rewrites and reloads the IPsec configuration for strongSwan.

svc

This script gives control over the services running on the firewall, similar to interacting with services at **Status > Services**.

The general form of the command is:

```
pfSsh.php 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
```

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*.

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, but is accessible from the console.

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.

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 removed access to the GUI.

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.

Time Synchronization

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 embedded systems, some of 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.

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 hardware found in the [pfSense 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
```

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](#).

Troubleshooting

The **Setup Wizard** and related configuration tasks will work for most situations, but there may be issues getting connections to flow normally in their intended directions. Some of these issues may be unique to a particular environment or configuration, but can be worked through with basic troubleshooting.

Cannot access WebGUI from LAN

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 (One 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 [What to do when locked out of the WebGUI](#). 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 pfSense 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`.

No Internet from LAN

If the client PC is able to reach the WebGUI but not the Internet, there are several things to consider. The WAN interface may not be properly configured, DNS resolution may not be working, there could be a problem with the firewall rules, NAT rules, or even something as simple as a local gateway issue.

WAN Interface Issues

First, check the WAN interface to be sure it is operational. Browse to **Status > Interfaces** and look at the **WAN** interface status there. If the interface is working properly the status will show as “up”. If it shows “no carrier” or “down”, double check the cabling and WAN settings under **Interfaces > WAN**.

If the interface is using PPPoE or PPTP for the WAN type, there is an additional status line indicating if the PPP connection is active. If it is down, try pressing the  **Connect** button. If that doesn't work, double check all of the interface settings on **Interfaces > WAN**, check or reboot the ISP CPE (cable/DSL modem, etc.), and perhaps consult with the ISP for help regarding the settings and circuit state.

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, a pfSense firewall 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 pfSense 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 pfSense firewall or an alternate set of working internal or external DNS servers.

Another possibility for DNS working from the pfSense 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.

Client Gateway Issue

In order for a pfSense firewall to properly route Internet traffic for client PCs, it must be their gateway. If client PCs are configured using the DHCP server built into pfSense firewalls, this will be set automatically. However, if the clients receive DHCP information from an alternate DHCP server, or their IP addresses have been entered manually, double check that their gateway is set for the IP address of the interface to which they connect on the pfSense firewall. For example, if the clients are on the pfSense LAN interface and the IP address for the LAN interface is 192.168.1.1, then the gateway address on the client PCs must be set to 192.168.1.1.

Firewall Rule Issues

If the default “LAN to Any” rule has been changed or removed from the LAN interface, traffic attempting to reach the Internet from client PCs via the pfSense firewall may be blocked. This is easily confirmed by browsing to **Status > System Logs** and looking at the **Firewall** tab. If there are entries there that show blocked connections from LAN PCs trying to reach Internet hosts, revisit the LAN ruleset at **Firewall > Rules**, on the **LAN** tab to make the necessary adjustments to allow that traffic. Consult [Firewall](#) for more detailed information on editing or creating additional rules.

If it works from the LAN side but not from an OPT interface, be sure the firewall has rules in place to allow the traffic to pass. Rule are not created by default on OPT interfaces.

NAT Rule Issues

If the outbound NAT rules have been changed from their defaults, traffic attempting to reach the Internet may not have NAT properly applied. Navigate to **Firewall > NAT, Outbound** tab. In most cases the setting should be on **Automatic outbound NAT rule generation**. If it is not, change to that setting, Click **Save** and **Apply Changes**, and then try to reach the Internet from a client PC again. If that did not help a PC on the LAN to get out, then the issue is likely elsewhere.

If Outbound NAT is set to **Manual Outbound NAT rule generation** and Internet access works from LAN but not from an OPT interface, manually add rules that matches traffic coming from the OPT interface. Look at the existing rules for LAN and adjust them accordingly, or refer to [Outbound NAT](#) for more information on creating outbound NAT rules.

The same applies for traffic coming from VPN users: OpenVPN, IPsec, etc. If these users need to reach the Internet via this pfSense firewall, they will need outbound NAT rules for their subnets. See [Outbound NAT](#) for more information.

pfSense XML Configuration File

pfSense firewalls store all of their settings in an XML format configuration file. All configuration settings including settings for packages are held in this one file. All other configuration files for system services and behavior are generated dynamically at run time based on the settings held within the 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 handles everything automatically.

Most people will never need to know where the configuration file resides, but for reference it is in `/cf/conf/config.xml`. Typically, `/conf/` is a symlink to `/cf/conf`, so it may also be accessible directly from `/conf/config.xml`, but this varies by platform and filesystem layout.

Manually editing the configuration

A few 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 book.

Warning: Even for seasoned administrators it is still easy to incorrectly edit the configuration file. Always keep backups and be aware that breaking the configuration will result in unintended consequences.

The safest and easiest method of editing the configuration file is to make a backup from **Diagnostics > Backup/Restore**, save the file to a PC, edit the file and make any needed changes, then restore the altered configuration file to the firewall. Use an 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.

For administrators familiar with the `vi` editor, the `vi config` command will edit the running configuration live, and after saving and quitting the editor, the firewall will remove the cached configuration from `/tmp/config.cache` and then the changes will be visible in the GUI. The changes will not be active until the next time the service relevant to the edited portion of the config is restarted/reloaded.

What to do when locked out of the WebGUI

Under certain circumstances an administrator can be locked out of the WebGUI. 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 credentials:

Username `admin`

Password `pfSense`

Forgotten Password

The firewall administrator password can easily be reset using the firewall console if it has been lost. Access the physical console (Serial or Keyboard/Monitor) and use option 3 to reset the WebGUI password. This option can also reset the admin account if it has been disabled or expired.

After resetting the password, the `admin` user can login with the default password `pfSense`.

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:

- Reboot the pfSense firewall
- Choose the *Boot Single User* option (2) from the loader menu (the one with the ASCII pfSense logo)
- Press `enter` when prompted to start `/bin/sh`
- Remount all partitions as rewritable:

```
# /sbin/mount -a -t ufs
```

- Run the built-in password reset command:

```
# /etc/rc.initial.password
```

- Follow the prompts to reset the password
- Reboot

When the firewall reboots, the `admin` user can login with the default password `pfSense`.

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://pfsensehostname:443`
- `https://pfsensehostname: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 WebGUI back to HTTP.

Blocked Access with Firewall Rules

If a remote administrator loses access to the WebGUI 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 WebGUI 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!

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 Internet-connected WAN interface. **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” on 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 WebGUI 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 WebGUI with that temporary fix, the administrator must perform whatever work is required in the WebGUI to make the fix permanent. When the rules are saved in the WebGUI, the temporary edit to `/tmp/rules.debug` will be overwritten.

Remotely Circumvent Firewall Lockout with SSH Tunneling

If remote access to the WebGUI is blocked by the firewall, but SSH access is allowed, then there is a relatively easy way to get in: SSH Tunneling.

If the WebGUI is on port 80, set the SSH client to forward local port 443 (or 4443, or another port) to remote port `localhost:443`. If the firewall WebGUI 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://`.

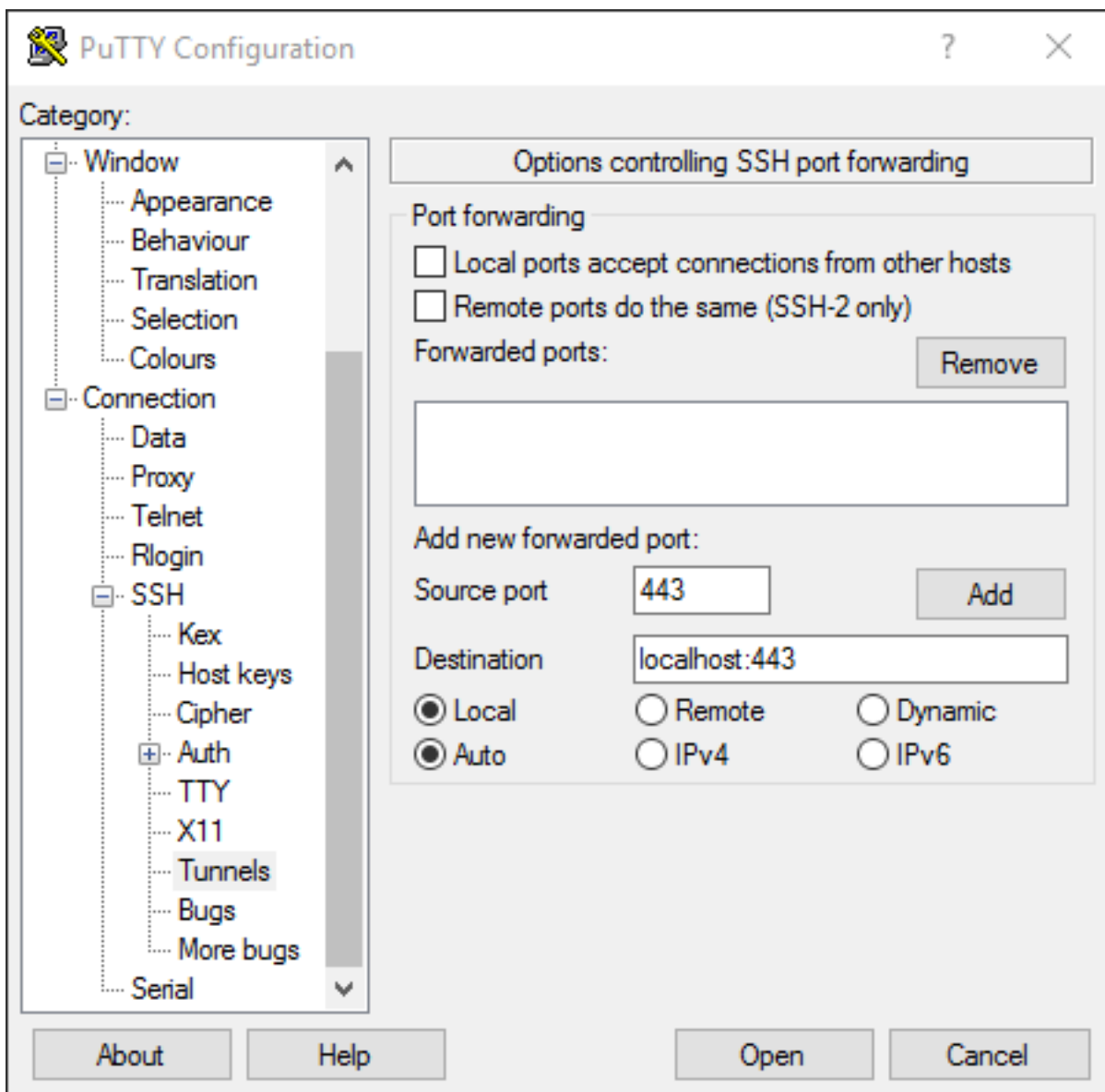


Fig. 7.16: Setting Up a Port 80 SSH Tunnel in PuTTY

Fill out the options as shown in Figure *Setting Up a Port 80 SSH Tunnel in PuTTY*, then click **Add**.

Once the client connects and authenticates, the WebGUI is accessible from the redirected local port.

Locked Out Due to Squid Configuration Error

If a firewall administrator accidentally configures Squid to use the same port as the WebGUI, 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 WebGUI wants, then the WebGUI will not be accessible to fix the configuration.

The following procedure may help to regain control.

- Connect to the pfSense 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 WebGUI process, and then attempt to access the WebGUI 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.

Most pfSense configuration is performed using the web-based GUI configurator (webConfigurator), or WebGUI for short. 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.

Connecting to the WebGUI

In order to reach the WebGUI, connect with a web browser from a computer connected to the LAN. 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 pfSense system is 192.168.1.1 with a /24 mask (255.255.255.0), and there is also a DHCP server running. If the computer is set to use DHCP, it should obtain an address in the LAN subnet automatically. Then open a browser and navigate to <https://192.168.1.1>.

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.

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 is disabled, client computers on LAN must have an IP address in the pfSense LAN subnet statically configured, such as 192.168.1.5, with a subnet mask that matches the one given to pfSense, such as 255.255.255.0.

Once the computer is connected to the same LAN as pfSense, navigate to the firewall LAN IP address. 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. To access the GUI directly without the redirect, use <https://192.168.1.1>.

When loading the WebGUI, the firewall first presents a login page. On this page, enter the default credentials:

username admin

password pfsense

INTERFACE TYPES AND CONFIGURATION

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.

To create an interface group:

- Navigate to **Interfaces > (assign), Interface Groups** tab
- Click  **Add** to create a new group
- Enter a **Group Name**. This name may only contain upper and lowercase letters, no numbers, spaces, or special characters
- Enter a **Group Description** (optional)
- Add interfaces as **Group Members** by ctrl-clicking to select entries from the interface list
- Click **Save**

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.

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.

Interface Group Configuration

Group Name	LocalNetworks No numbers or spaces are allowed. Only characters: a-zA-Z
Group Description	Local interfaces/networks A group description may be entered here for administrative reference (not parsed).
Group Members	WAN LAN DMZ WAN2 NOTE: Rules for WAN type interfaces in groups do not contain the reply-to mechanism upon which Multi-WAN typically relies. More Information

 Save

Fig. 8.1: Add Interface Group

Firewall / Rules / LocalNetworks

Floating

LocalNetworks

WAN

LAN

DMZ

WAN2

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination

Fig. 8.2: 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 overridden 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.

Use with WAN Interfaces

We do not recommend using 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 (Gateway selected on the interface configuration) 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.

Wireless

The **Wireless** tab under **Interfaces > (assign)** is for creating and managing additional Virtual Access Point (VAP) interfaces. Using VAPs allows multiple networks with unique wireless SSIDs to be run off a single card, if that feature is supported by the hardware and driver in use. A VAP is created on the **Wireless** tab, then assigned on the **Interface assignments** tab. In-depth information on this feature can be found in [Wireless](#).

VLANs

VLAN tagged interfaces, or 802.1Q tagged interfaces, are located on the **VLANs** tab under **Interfaces > (assign)**. VLAN instances allow the system to address traffic tagged by an 802.1Q capable switch separately as if each tag were its own distinct interface. A VLAN is created on this tab, then assigned on the **Interface assignments** tab. In-depth information on this feature can be found in [Virtual LANs \(VLANs\)](#).

QinQs

The QinQs tab under **Interfaces > (assign)** allows creating an 802.1ad compatible interface that is also known as *Stacked VLANs*. This feature allows multiple VLAN tags to be contained in a single packet. This can aid in carrying VLAN-tagged traffic for other networks across an intermediate network using a different or overlapping tag. In-depth information on this feature can be found in [pfSense QinQ Configuration](#).

Bridges

Interface Bridges, or multiple interfaces tied together into a common shared layer 2 broadcast domain, are created and managed on the **Bridges** tab under **Interfaces > (assign)**. More information on bridging, including how to create and manage bridges, is in [Bridging](#).

OpenVPN

After An OpenVPN instance is created, it may be assigned under **Interfaces > (assign)**. Assigning an OpenVPN interface enables interface-specific rules, and allows the interface to be used elsewhere in the GUI that requires an assigned interface. This also triggers the creation of a dynamic gateway. This gateway can be used for policy routing, or in a gateway group for Multi-WAN. See [Assigning OpenVPN Interfaces](#) for more information.

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 > (assign)** on the **PPPs** tab.

Multi-Link PPP (MLPPP)

Editing a PPP instance also allows Multi-Link PPP (MLPPP) to be configured for supported providers. 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](#).

PPP (Point-to-Point Protocol) Interface Types

Add or edit a PPP entry as follows:

- Navigate to **Interfaces > (assign)** on the **PPPs** tab
- Click  to edit an existing entry or  to add a new entry
- Set the Link Type, which changes the remaining options on the page. The link types are explained throughout the remainder of this section.

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. Upon selecting the **PPP** link type, the **Link Interface(s)** list is populated with serial devices that can be used to communicate with a modem. Click on a specific entry to select it for use. After selecting the interface, optionally enter a **Description** for the PPP entry.

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.

When configuring a 3G/4G network, the **Service Provider** options pre-fill other relevant fields on the page.

- Select a **Country**, such as *United States*, to activate the **Provider** drop-down with known cellular providers in that country
- Select a **Provider** from the list, such as *T-Mobile*, to activate the **Plan** drop-down.
- Select a **Plan** and the remaining fields will be filled with known values for that **Provider** and **Plan**

The **Service Provider** options can be configured manually if other values are needed, or when using a provider that is not listed:

Username and Password The credentials used for the PPP login.

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.

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 identically to PPTP. Refer to the previous sections for configuration information.

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 it will immediately attempt to reconnect when a link is lost. This behavior is described as **Always On**. **Dial-on-Demand** will delay 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: pfSense will perform gateway monitoring by default which will generate 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 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. The option should almost always remain enabled. 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**.

Short Sequence (ShortSeq) This option is only meaningful if MLPPP is negotiated. 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 (AFCComp) 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**.

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. IPsec in **transport** mode can use GRE for tunneling encrypted traffic in a way that allows for traditional routing or the use of routing protocols. The GRE protocol was originally designed by Cisco, and it is the default tunneling mode on many of their devices.

To create or manage a GRE interface:

- Navigate to **Interfaces > (assign), GRE** tab
- Click  **Add** to create a new GRE instance, or click  to edit an existing interface.
- Complete the settings as follows:
 - Parent interface** The interface upon which the GRE tunnel will terminate. Often this will be WAN or a WAN-type connection.
 - GRE 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.
 - GRE 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.
 - GRE tunnel remote address** The address used by the firewall inside the tunnel to reach the other end. Traffic destined for the other end of the tunnel must use this address as a gateway for routing purposes.
 - GRE Tunnel Subnet** The subnet mask for the GRE interface address.
 - Description** A short description of this GRE tunnel for documentation purposes.
- Click **Save**

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 tunnel brokers such as [Hurricane Electric](#) and [SixXS](#) in locations where IPv6 connectivity is unavailable.

See also:

See [Connecting with a Tunnel Broker Service](#) for information about connecting to a tunnelbroker 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.

To create or manage a GIF interface:

- Navigate to **Interfaces > (assign), GIF** tab

- Click  **Add** to create a new GIF instance, or click  to edit an existing interface.
- Complete the settings as follows:

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 other end. 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. In this example it would be 64.

Route Caching The Route caching option controls whether or not the route to the remote endpoint is cached. If the path to the remote peer is static, setting this can avoid one route lookup per packet. However if the path to the far side can change, this option could result in the GIF traffic failing to flow when the route changes.

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.

Description A short description of this GIF tunnel for documentation purposes.

- Click **Save**

Note: If the GIF interface is assigned under **Interfaces > (assign)**, set the **IPv4 Configuration Type** and **IPv6 Configuration Type** to *None*. The firewall will automatically create a dynamic gateway in this situation.

LAGG (Link Aggregation)

Link aggregation is handled by `lagg(4)` type interfaces (LAGG) on pfSense. 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.

To create or manage LAGG interfaces:

- Navigate to **Interfaces > (assign)**, **LAGGs** tab
- Click  **Add** to create a new LAGG, or click  to edit an existing instance.

- Complete the settings as follows:

Parent Interfaces This list contains all currently unassigned interfaces, and 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 There are currently six different operating modes for LAGG interfaces: LACP, Failover, Load Balance, FEC, 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 known 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 pfSense, 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.

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. The primary interface is the first interface selected in the list, and will continue in order until it reaches the end of the selected interfaces.

Note: By default, traffic may only be received on 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.

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.

FEC FEC mode supports Cisco EtherChannel and is an alias for Load Balance mode.

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.

- Click **Save**

After creating a LAGG interface, it works like any other physical interface. Assign the lagg interface under **Interfaces > (assign)** and give it an IP address, or build other things on top of it such as VLANs.

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.

Interface Configuration

To assign a new interface:

- Navigate to **Interfaces > (assign)**
- Pick the new interface from the **Available network ports** list
- Click  **Add**

The newly assign 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.

The following options are available for all interface types.

Description

The name of the interface. This will change the name of the interface on the **Interfaces** menu, on the tabs under **Firewall > Rules**, under **Services > DHCP**, and elsewhere throughout the GUI. Interface names may only contain letters, numbers and the only special character that is allowed is an underscore (“_”). 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 WAN Types](#).

IPv6 Configuration Type

Configures the IPv6 settings for the interface. Details for this option are in [IPv6 WAN Types](#).

MAC address

The MAC address of an interface can be changed (“spoofed”) to mimic a previous piece of equipment.

Warning: We recommend avoiding this practice. The old MAC would 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. It 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.

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. ARP cache problems tend to be very temporary, resolving automatically within minutes or by power cycling other equipment.

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.

Block Private Networks

When **Block private networks** is active pfSense inserts a rule automatically that 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 usually 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 pfSense will block traffic from a list of unallocated and reserved networks. This list is periodically updated by the firewall automatically.

Now that the IPv4 space has all been assigned, this list is quite small, containing mostly networks that have been reserved in some way by IANA. These subnets should never be in active use on a network, especially one facing the Internet, so it's a good practice to enable this option on WAN type interfaces. For IPv6, the list is quite large,

containing sizable chunks of the possible IPv6 space that has yet to be allocated. On systems with low amounts of RAM, this list may be too large, or the default value of **Firewall Maximum Table Entries** may be too small. That value may be adjusted under **System > Advanced** on the **Firewall & NAT** tab.

IPv4 WAN Types

Once an interface has been assigned, in most cases it will require an IP address. For IPv4 connections, the following choices are available: Static IPv4, DHCP, PPP, PPPoE, PPTP, and L2TP. These options are selected using the **IPv4 Configuration Type** selector on an interface page (e.g. **Interfaces > WAN**).

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 IP address on the interface is being managed in some other way, such as for an OpenVPN or GIF interface.

Static IPv4

With **Static IPv4**, the interface contains a manually configured IP address. When chosen, three additional fields are available on the interface configuration screen: **IPv4 Address**, a CIDR subnet mask selector, and the **IPv4 Upstream Gateway** field.

To configure the interface for static IPv4 on an internal interface (e.g. LAN, DMZ):

- Select *Static IPv4* under **IPv4 Configuration Type**
- Enter the IPv4 address for the interface into the **IPv4 address** box
- Choose the appropriate subnet mask from the CIDR drop-down after the address box
- Do not select an **IPv4 Upstream Gateway**

To configure the interface for static IPv4 on a WAN type interface:

- Select *Static IPv4* under **IPv4 Configuration Type**
- Enter the IPv4 address for the interface into the **IPv4 address** box
- Choose the appropriate subnet mask from the CIDR drop-down after the address box
- Perform one of the following to use a gateway on the interface:
 - Select an **IPv4 Upstream Gateway** from the list, *OR*
 - Click  **Add a new gateway** 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 circuit. 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, the Quality Graphs, and elsewhere.

Gateway IPv4 The IP address of the gateway. This address must be inside of the same subnet as the Static IPv4 address when using this form.

Description A bit of text to indicate the purpose of the gateway.

* Click  Add

Note: Selecting an **IPv4 Upstream Gateway** from the drop-down list or adding and selecting a new gateway will make pfSense treat this interface as a WAN type interface for NAT and related functions. This is not desirable for internal interfaces such as LAN or a DMZ. 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.

DHCP

When an interface is set to **DHCP**, pfSense 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 an IP address supplied by the specified server.

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 in the pfSense GUI.

PPP Types

The various PPP-based connection types such as PPP, PPPoE, PPTP, and L2TP are all covered in detail earlier in this chapter ([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 > (assign)** on the **PPPs** tab, find the entry, and edit it there.

IPv6 WAN 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.

See also:

For more information on IPv6, including a basic introduction, see [IPv6](#).

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 IP address on the interface is being managed in some other way, such as for an OpenVPN or GIF interface.

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 circuit. Changing the default IPv4 gateway has no effect on the IPv6 gateway, and vice versa.

DHCP6

DHCP6 configures pfSense to attempt automatic IPv6 configuration of this interface via DHCPv6. DHCPv6 will configure the interface with an IP 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 pfSense. 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](#). 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.

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 GUI.

SLAAC

Stateless address autoconfiguration (*SLAAC*) as the IPv6 type makes pfSense attempt to configure the IPv6 address for the interface from router advertisements (RA) that advertise the prefix and related information. Note that DNS is not typically provided via RA, so pfSense will still attempt to get the DNS servers via DHCPv6 when using SLAAC. In the future, the RDNSS extensions to the RA process may allow DNS servers to be obtained from RA. For more information on router advertisements, see [Router Advertisements](#).

6RD Tunnel

6RD is an IPv6 tunneling technology employed by some ISPs to quickly enable IPv6 support for their networks, passing IPv6 traffic inside specially crafted IPv4 packets between an 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.

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.

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*.

pfSense 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 > (assign)**.

Physical and Virtual Interfaces

Most interfaces discussed in this chapter can be assigned as WAN, LAN, or an OPT interface under **Interfaces > (assign)**. All currently-defined and detected interfaces are listed directly on **Interfaces > (assign)** or in the list of interfaces available for assignment. By default, this list includes only the physical interfaces, but the other tabs under **Interfaces > (assign)** can create virtual interfaces which can then be assigned.

Interfaces on pfSense support various combinations of options on the interfaces themselves. 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.

USER MANAGEMENT AND AUTHENTICATION

User Management

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.

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, when editing the existing user or group, click  **Add** in the **Assigned Privileges** or **Effective Privileges** section.

A list of all available privileges is presented. Privileges may be added one at a time by selecting a single entry, or by multi-select using ctrl-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 Lets the user access any page in the GUI

WebCfg - Dashboard (all) Lets the user access 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 Does not allow the user to make changes to the firewall config (*config.xml*). Note that this does not prevent the user from taking other actions that do not involve writing to the config.

User - System - Shell account access Gives 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, they will be forwarded to the first page in their privilege list which they have permission to 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.

Adding/Editing Users

The **Users** tab under **System > User Manager** is where individual users are managed. To add a new user, click .

Add, to edit an existing user, click .

Before permissions may be added to a user, it must first be created, so the first step is always to add the user and save. If multiple users need the same permissions, it is easier to add a group and then add users to the group.

To add a user, click  **Add** and the new user screen will appear.

Disabled This checkbox controls whether this user will be active. If this account should be deactivated, check this box.

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 and **Confirmation** are also required. Passwords are stored in the pfSense configuration as hashes. Ensure the two fields match to confirm the password.

Full Name Optional field which can be used to enter a longer name or a description for a user account.

Expiration Date May also be defined if desired to deactivate the user automatically when that date has been reached. The date must be entered in *MM/DD/YYYY* format.

Group Memberships If groups have already been defined, this control may be used to add the user as a member. To add a group for this user, find it in the **Not Member Of** column, select it, and click  to move it to the **Member Of** column. To remove a user from the group, select it from the **Member**

Of column and click  to move it to the **Not Member Of** column.

Effective Privileges Appears when editing an existing user, not when adding a user. See [Privileges](#) for information on managing privileges. If the user is part of a group, the group's permissions are shown in this list but those permissions cannot be edited, however additional permissions may be added.

Certificate Behavior of this section changes depending on whether a user is being added or edited. When adding a user, to create a certificate check **Click to create a user certificate** to show the form to create a certificate. Fill in the **Descriptive name**, choose a **Certificate Authority**, select a **Key Length**, and enter a **Lifetime**. For more information on these parameters, see [Create an Internal Certificate](#). If editing a user, this section of the page instead becomes a list of user certificates. From

here, click  **Add** to add a certificate to the user. The settings on that page are identical to [Create an Internal Certificate](#) except even more of the data is pre-filled with the username. If the certificate already exists, select *Choose an Existing Certificate* and then pick an **Existing Certificate** from the list.

Authorized keys SSH public keys may be entered for shell or other SSH access. To add a key, paste or enter in the key data.

IPsec Pre-Shared Key Used for a non-xauth Pre-Shared Key mobile IPsec setup. If an **IPsec Pre-Shared Key** is entered here, the username is used as the identifier. The PSK is also displayed under **VPN > IPsec** on the **Pre-Shared Keys** tab. If mobile IPsec will only be used with xauth, this field may be left blank.

After saving the user, click  on the user's row to edit the entry if necessary.

Adding/Editing Groups

Groups are a great way to manage sets of permissions to give users so that they do not need to be maintained individually on every user account. For example, a group could be used for IPsec xauth users, or a group that can access the firewall's dashboard, a group of firewall administrators, or many other possible scenarios using any combination of privileges.

As with users, a group must first be created before privileges can be added. After saving the group, edit the group to add privileges.

Groups are managed under **System > User Manager** on the **Groups** tab. To add a new group from this screen, click



Add. To edit an existing group, click  next to its entry in the list.

..note:: When working with LDAP and RADIUS, local groups must exist to match the groups the users are members of on the server. For example, if an LDAP group named “firewall_admins” exists then pfSense must also contain a group named identically, “firewall_admins”, with the desired privileges. Remote groups with long names or names containing spaces or other special characters must be configured for a *Remote Scope*.

Start the process of adding a group by clicking  **Add** and the screen to add a new group will appear.

Group name 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. This can feel somewhat limited when working with groups from LDAP, for example, but usually it's easier to create or rename an appropriately-named group on the authentication server instead of attempting to make the firewall group match.

Scope Can be set *Local* for groups on the firewall itself (such as those for use in the shell), or *Remote* to relax the group name restrictions and to prevent the group name from being exposed to the base operating system. For example, *Remote* scope 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, find it in the **Not Members** column, select it, and click  to move it to the **Members** column. To remove a user from the group, select it from the **Members** column and click  to move it to the **Not Members** column.

Assigned Privileges Appears only when editing an existing group. This section allows adding privileges to the group. See *Privileges* earlier in this for information on managing privileges.

Settings

The **Settings** tab in the User Manager controls two things: How long a login session is valid, and where the GUI logins will prefer to be authenticated.

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 make it 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*. If the RADIUS or LDAP server is unreachable for some reason, the authentication will fall back to *Local Database* even if another method is chosen.

When using a RADIUS or LDAP server, the users and/or group memberships must still be defined in the firewall in order to properly allocate permissions, as there is not yet a method to obtain permissions dynamically from an authentication server.

For group membership to work properly, pfSense must be able to recognize the groups as presented by the authentication server. This requires two things:

1. The local groups must exist with identical names.
2. pfSense 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.

Authentication Servers

Using the **Authentication Servers** tab under **System > User Manager**, RADIUS and LDAP servers may be defined as authentication sources. See [Support Throughout pfSense](#) for information on where these servers may be used in

pfSense currently. To add a new server from this screen, click  **Add**. To edit an existing server, click  next to its entry.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the August 2015 Hangout on RADIUS and LDAP.

RADIUS

To add a new RADIUS server:

- Make sure that the RADIUS server has the firewall defined as a client before proceeding.
- Navigate to **System > User Manager, Authentication Servers** tab.
- Click  **Add**.
- Set the **Type** selector to *RADIUS*. The RADIUS Server Settings will be displayed.
- Fill in the fields as described below:

Descriptive name The name for this RADIUS server. This name will be used to identify the server throughout the pfSense GUI.

Hostname or IP address The address of the RADIUS server. This can be a fully qualified domain name, or an IPv4 IP address.

Shared Secret The password established for this firewall *on the RADIUS server* software.

Services offered This selector set which services are offered by this RADIUS server. *Authentication and Accounting*, *Authentication only*, or *Accounting only*. *Authentication* will use this RADIUS server to authenticate users. *Accounting* will send RADIUS start/stop accounting packet data for login sessions if supported in the area where it is used.

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.

- Click **Save** to create the server.
- Visit **Diagnostics > Authentication** to test the RADIUS server using a valid account.

For RADIUS groups, the RADIUS server must return a list of groups in the `Class` RADIUS reply attribute as a string. Multiple groups must be separated by a semicolon.

For example, in FreeRADIUS, to return the “admins” and “VPNUsers” groups, the following Reply-Item RADIUS Attribute would be used:

```
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, ensure they exist on pfSense with matching names and that the server is returning the `Class` attribute as a string, not binary.

LDAP

To add a new LDAP server:

- Make sure that the LDAP server can be reached by the firewall.
- If SSL will be used, import the Certificate Authority used by the LDAP server into pfSense before proceeding. See [Certificate Authority Management](#) for more information on creating or importing CAs.
- Navigate to **System > User Manager, Servers** tab.
- Click  **Add**.
- Set the **Type** selector to *LDAP*. The LDAP Server Settings will be displayed.
- Fill in the fields as described below:

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 SSL will be used, a Hostname must be specified here and the hostname **must** match the **Common Name** of the server certificate presented by the LDAP server and that hostname must resolve to the IP address of the LDAP server, e.g. *CN=ldap.example.com*, and *ldap.example.com* is *192.168.1.5*. The only exception to this is if the IP address of the server also happens to be the CN of its server certificate.

This can be worked around in some cases by creating a DNS Forwarder host override to make the server certificate CN 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 TCP port is 389, 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, pfSense uses an *ldaps://* URL, it does not support 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 to communicate with the LDAP server. The first, and default, selection is *TCP - Standard* which uses plain TCP connections on port 389. A more secure choice, if the LDAP server supports it, is *SSL - Encrypted* on port 636. The SSL choice will encrypt the LDAP queries made to the server, which is especially important if the LDAP server is not on a local network segment.

Note: It is always recommended to use SSL where possible, though plain TCP is easier to setup and diagnose since a packet capture would show the contents of the queries and responses.

Peer Certificate Authority If *SSL - Encrypted* was chosen for the **Transport**, then the value of this selector is used to validate the certificate of the LDAP server. The selected CA must match the CA configured on the LDAP server, otherwise problems will arise. See [Certificate Authority Management](#) for more information on creating or importing CAs.

Protocol version Chooses which version of the LDAP protocol is employed by the LDAP server, either 2 or 3, typically 3.

Search scope Determines where, and how deep, a search will go for a match.

Level Choose between *One Level* or *Entire Subtree* to control how deep the search will go. *Entire Subtree* is the best choice when the decision is not certain, 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 semicolon-separated list of potential account locations or containers. These containers will be prepended to the search Base DN above or specify a full container path here and leave the Base DN blank. If the LDAP server supports it, and the bind settings are correct, click the **Select** button to browse the LDAP server containers and select them there. Some examples of these containers are:

- `CN=Users;DC=example;DC=com` This would search for users inside of the domain component *example.com*, a common syntax to see for Active Directory
- `CN=Users,DC=example,DC=com;OU=OtherUsers,DC=example,DC=com` This would search 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. To set an Extended Query, check the box and fill in the 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. By default the **Use anonymous binds to resolve distinguished names** box is checked to perform an anonymous bind. If the server requires authentication to bind and perform a query, uncheck that box and then specify a **User DN** and **Password** to be used for the bind.

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.

Initial Template 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 a user's name, 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 unchecked, Active Directory style group membership is used where groups are listed as an attribute of the user object. When checked, RFC 2307 style group membership is used where the users are listed as members on the group object.

Note: When this is used, the **Group member attribute** may also need changed, typically it would be set to *memberUid* in this case, but may vary by LDAP schema.

Group Object Class Used with RFC 2307 style groups, it specifies the object class of the group, typically *posixGroup* but it may vary by LDAP schema. It is not needed for Active Directory style groups.

UTF8 Encode When checked, queries to the LDAP server will be UTF8-encoded and the responses will be UTF8-decoded. Support varies depending on the LDAP server. Generally only necessary if user names, groups, passwords, and other attributes contain non-traditional 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.

- Click **Save** to create the server.
- Visit **Diagnostics > Authentication** to test the LDAP server using a valid account.

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 on pfSense with matching names and that the proper group structure is selected (e.g. RFC 2703 groups may need to be selected.)

External 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.

RADIUS Server Example

This example was made against FreeRADIUS but doing the same for Windows Server would be identical. See [RADIUS Authentication with Windows Server](#) 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`

OpenLDAP Example

In this example, pfSense is setup to connect 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*

Active Directory LDAP Example

In this example, pfSense is setup to connect 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 in pfSense, SSL may be used as well by selecting that option and choosing the appropriate CA from the **Peer Certificate Authority** drop down, and setting the **Hostname** to the common name of the server certificate.

Troubleshooting

Testing authentication servers is possible using the tool located at **Diagnostics > Authentication**. From that page, testing a user is simple:

- Navigate to **Diagnostics > Authentication**
- Select an **Authentication Server**
- Enter a **Username**
- Enter a **Password**
- Click the **Test** button.

The firewall will attempt to authenticate the given user against the specified server and will return the result. It is usually best 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 groups are printed in the test results.

If an error is received when testing authentication, double check the credentials and the server settings, then make any necessary adjustments and try again.

Active Directory LDAP Errors

The most common mistake with LDAP access to Active Directory is not specifying a proper bind user in the correct format. If the username alone does not work, enter the full Distinguished Name (DN) for the bind user, such as *CN=binduser,CN=Users,DC=domain,DC=local*.

If the full DN of the user is unknown, it can be found by navigating to the user in **ADSI Edit** found under **Administrative Tools** on the Windows Server.

Another common mistake with group membership is not specifying *Entire Subtree* for the **Search Scope Level**.

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 be used to determine what the DN for a given group will be.

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=)

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 on pfSense (**Status > System Logs**) may also contain some detail that hints at a resolution.

Troubleshooting via Packet Captures

Packet captures can be invaluable for diagnosing errors as well. If an unencrypted method (RADIUS, LDAP without SSL) 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*. For more information on packet captures, see [Packet Capturing](#).

0.230	TCP	31918 > 7dap [ACK] Seq=1 Ack=1 win=66608
0.230	LDAP	bindRequest(1) "Administrator" simple
0.243	LDAP	bindResponse(1) invalidCredentials (80090)
0.230	TCP	31918 > 7dap [ACK] Seq=31 Ack=111 win=664
0.230	LDAP	unbindRequest(2)
0.230	TCP	31918 > 7dap [FIN, ACK] Seq=38 Ack=111 win=0

Fig. 9.1: Sample LDAP Failure Capture

The User Manager in pfSense provides the ability to create and manage multiple user accounts. These accounts can be used to access the GUI, use VPN services like OpenVPN and IPsec, and use the Captive Portal.

The User Manager can also be used to define external authentication sources such as RADIUS and LDAP.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the February 2015 Hangout on User Management and Privileges, and the August 2015 Hangout on RADIUS and LDAP.

Support Throughout pfSense

As of this writing, not all areas of pfSense hook back into the User Manager.

pfSense 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.

OpenVPN Supports users in the User Manager, RADIUS or LDAP via User Manager.

IPsec Supports users in the User Manager, RADIUS or LDAP via User Manager for Xauth, and RADIUS for IKEv2 with EAP-RADIUS.

Captive Portal Support local users in the User Manager, and RADIUS users via settings in the Captive Portal page.

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.

CERTIFICATE MANAGEMENT

Certificate Authority Management

Certificate Authorities (CAs) are managed from **System > Cert Manager**, on the **CAs** tab. From this screen CAs may be added, edited, exported, or deleted.

Create a new Certificate Authority

To create a new CA, start the process as follows:

- Navigate to **System > Cert Manager** on the **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. These options and further instructions are in the corresponding sections below:
 - Create an Internal Certificate Authority
 - Import an Existing Certificate Authority
 - Create an Intermediate Certificate Authority

Create an Internal Certificate Authority

The most common **Method** used from here is to *Create an Internal Certificate Authority*. This will make a new root CA based on information entered on this page.

- Select the **Key length** to choose how “strong” the CA is in terms of encryption. The longer the key, the more secure it is. However, longer keys can take more CPU time to process, so it isn’t always wise to use the maximum value. The default value of *2048* is a good balance.
- Select a **Digest Algorithm** from the supplied list. The current best practice is to use an algorithm stronger than SHA1 where possible. *SHA256* is a good choice.

Note: Some older or less sophisticated equipment, such as VPN-enabled VoIP handsets may only support SHA1 for the Digest Algorithm. Consult device documentation for specifics.

- Enter a value for **Lifetime** to specify the number of days for which the CA will be valid. The duration depends on personal preferences and site policies. Changing the CA frequently is more secure, but it is also a management headache as it would require reissuing new certificates when the CA expires. By default the GUI suggests using 3650 days, which is approximately 10 years.

- Enter values for the **Distinguished name** section for personalized parameters in the CA. 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. Punctuation and special characters must not be used.
 - Select the **Country Code** from the list. This is the ISO-recognized country code, not a hostname top-level domain.
 - Enter the **State or Province** fully spelled out, not abbreviated.
 - Enter the **City**.
 - Enter the **Organization** name, typically the company name.
 - Enter a valid **Email Address**.
 - Enter the **Common Name (CN)**. This field is the internal name that identifies the CA. Unlike a certificate, the CN for a CA does not need to be the hostname, or anything specific. For instance, it could be called *VPNCA* or *MyCA*.

Note: Although it is technically valid, avoid using spaces in the CN.

- Click **Save**

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.

Import an Existing Certificate Authority

If an existing CA from an external source needs to be imported, it can be done by selecting the **Method** of *Import an Existing Certificate Authority*. 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.

- Enter the **Certificate data** for the CA. To trust a CA from another source, only the Certificate data for the CA is required. It 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-----
```

- Enter the **Certificate Private Key** if importing a custom external CA, or a CA that is capable of generating its own certificates and certificate revocation lists. This 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-----
```

- Enter the **Serial** for next certificate if the private key was entered. This is essential. A CA will create certificates each with a unique serial number in sequence. This value controls what the serial will be for the next certificate generated from this CA. 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.
- Click **Save**

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.

Importing a Chained or Nested Certificate Authority

If the CA has been signed by an intermediary and not directly by a root CA, it may be necessary to import both the root and the intermediate CA together in one entry, such as:

```
-----BEGIN CERTIFICATE-----
[Subordinate/Intermediate CA certificate text]
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
[Root CA certificate text]
-----END CERTIFICATE-----
```

Create an Intermediate Certificate Authority

An Intermediate CA will create a new CA that is capable of generating certificates, yet depends on another CA higher above it. To create one, select *Create an Intermediate Certificate Authority* from the **Method** drop-down.

Note: The higher-level CA must already exist on pfSense (Created or imported)

- Choose the higher-level CA to sign this CA using the **Signing Certificate Authority** drop-down. Only CAs with private keys present will be shown, as this is required to properly sign this new CA.
- Fill in the remaining parameters identical to those for *Create an Internal Certificate Authority*.

Edit a Certificate Authority

After a CA has been added, it can be edited from the list of CAs found at **System > Cert Manager** on the **CAs** tab.

To edit a CA, click the  icon at the end of its row. The screen presented allows editing the fields as if the CA were being imported.

For information on the fields on this screen, see *Import an Existing Certificate Authority*. In most cases the purpose of this screen would be to 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.

Export a Certificate Authority

From the list of CAs at **System > Cert Manager** on the **CAs** tab, the certificate and/or private key for a CA can be exported. 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.

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.

To export the certificate for a CA, click the  icon on the *left*. To export the private key for the CA, click the  icon on the *right*. Hover the mouse pointer over the icon and a tooltip will display the action to be performed for easy confirmation. The files will download with the descriptive name of the CA as the file name, with the extension *.crt* for the certificate, and *.key* for the private key.

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.
- Remove entries utilizing the CA or select a different CA.
- Navigate to **System > Cert Manager** on the **CAs** tab.
- Find the CA to delete in the list.
- Click  at the end of the row for the CA.
- Click OK on the confirmation dialog.

If an error appears, follow the on-screen instructions to correct the problem and then try again.

Certificate Management

Certificates are managed from **System > Cert Manager**, on the **Certificates** tab. From this screen Certificates may be added, edited, exported, or deleted.

Create a new Certificate

To create a new certificate, start the process as follows:

- Navigate to **System > Cert Manager** on the **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:
 - Import an Existing Certificate
 - Create an Internal Certificate
 - Create a Certificate Signing Request

Import an Existing Certificate

If an existing certificate from an external source needs to be imported, it can be done by selecting the **Method** of *Import an Existing Certificate*. This can be useful for certificates that have been made using another system or for certificates that have been provided by a third party.

- Enter the **Certificate data**, this is required. It is typically contained in a file ending with *.crt*. It would be plain text, and enclosed in a block such as:

```
-----BEGIN CERTIFICATE-----  
[A bunch of random-looking base64-encoded data]  
-----END CERTIFICATE-----
```

- Enter the **Private key data** which is also required. This 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-----
```

- Click Save to finish the import process.

If any errors are encountered, follow the on-screen instructions to resolve them. 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.

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** by which this certificate will be signed. 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.
- Select the **Key length** to choose how “strong” the certificate is in terms of encryption. The longer the key, the more secure it is. However, longer keys can take more CPU time to process, so it isn’t always wise to use the maximum value. The default value of 2048 is a good balance.
- Select a **Digest Algorithm** from the supplied list. The current best practice is to use an algorithm stronger than SHA1 where possible. *SHA256* is a good choice.

Note: Some older or less sophisticated equipment, such as VPN-enabled VoIP handsets may only support SHA1 for the Digest Algorithm. Consult device documentation for specifics.

- Select a **Certificate Type** which matches the purpose of this certificate.
 - Choose **Server Certificate** if the certificate will be used in a VPN server or HTTPS server. This indicates inside the certificate that it may be used in a server role, and no other.

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 signify 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 a constraint indicating that they are not a CA, and have nsCertType set to “server”.

- Choose **User Certificate** if the certificate can be used in an end-user capacity, such as a VPN client, but it cannot be used as a server. This prevents a user from using their own certificate to impersonate a server.

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.

- Choose **Certificate Authority** to create an intermediate CA. A certificate generated in this way will be subordinate to the chosen CA. It can create its own certificates, but the root CA must also be included when it is used. This is also known as “chaining”.
- Enter a value for **Lifetime** to specify the number of days for which the certificate will be valid. The duration depends on personal preferences and site policies. Changing the certificate frequently is more secure, but it is also a management headache as it requires reissuing new certificates when they expire. By default the GUI suggests using 3650 days, which is approximately 10 years.
- Enter values for the **Distinguished name** section for personalized parameters in the certificate. Most of these fields will be pre-populated with data from the CA. 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 certificate, and to distinguish one certificate from another. Punctuation and special characters must not be used.

- Select the **Country Code** from the list. This is the ISO-recognized country code, not a hostname top-level domain.
- Enter the **State or Province** fully spelled out, not abbreviated.
- Enter the **City**.
- Enter the **Organization** name, typically the company name.
- Enter a valid **Email Address**.
- Enter the **Common Name** (CN). This field is the internal name that identifies the certificate. Unlike a CA, the CN for a certificate should be a username or hostname. For instance, it could be called *VPNCert*, *user01*, or *vpnrouter.example.com*.

Note: Although it is technically valid, avoid using spaces in the CN.

- Click  **Add** to add **Alternative Names** if they are required. Alternative Names allow the certificate to specify multiple names that are all valid for the CN, such as two different hostnames, an additional IP address, a URL, or an e-mail address. This field may be left blank if it is not required or its purpose is unclear.
 - Enter a **Type** for the Alternative Name. This must contain one of *DNS* (FQDN or Hostname), *IP* (IP address), *URI*, or *email*.
 - Enter a **Value** for the Alternative Name. This field must contain an appropriately formatted value based on the Type entered.
 - Click  **Delete** at the end of the row for an unneeded Alternative Name.
 - Repeat this process for each additional **Alternative Name**.
- Click **Save**.

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.

Create a Certificate Signing Request

Choosing a **Method** of *Certificate Signing Request* creates a new request file that can be sent into a third party CA to be signed. This would be used to obtain a certificate from a trusted root certificate authority. Once this **Method** has been chosen, the remaining parameters for creating this certificate are identical to those for *Create an Internal Certificate*.

Export a Certificate

From the list of certificates at **System > Cert Manager** on the **Certificates** tab, a certificate and/or its private key may be exported.

To export the certificate, click the  icon. To export the private key for the certificate, click the  icon. To export the CA certificate, certificate and the private key for the certificate together in a PKCS#12 file, click the  icon. To confirm the proper file is being exported, hover the mouse pointer over the icon and a tooltip will display the action to be performed.

The files will download with the descriptive name of the certificate as the file name, and the extension `.crt` for the certificate and `.key` for the private key, or `.p12` for a PKCS#12 file.

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.
- 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.

If an error appears, follow the on-screen instructions to correct the problem and then try again.

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 be the user's username.

Local Authentication / Create Certificate When Creating a User

If user authentication is being performed on pfSense, the user certificate can be made inside of the User Manager.

- Navigate to **System > User Manager**
- Create a user. See [User Management and Authentication](#) for details.
- Fill in the **Username** and **Password**
- Select **Click to create a user certificate** in the User Certificates section, which will display a simple form for creating a user certificate.
 - Enter a short **Descriptive Name**, which can be the username or something such as *Bob's Remote Access VPN Cert*.
 - Choose the proper **Certificate Authority** for the VPN.
 - Adjust the **Key Length** and **Lifetime** if desired.
- Finish any other required user details.
- Click **Save**

Local Authentication / Add a Certificate to an Existing User

To add a certificate to an existing user:

- Navigate to **System > User Manager**
- Click  to edit the user
- Click  **Add** under User Certificates.
- Choose options as needed available from the certificate creation process described in *Create a new Certificate*, or select *Choose an existing certificate* and then select from the **Existing Certificates**

For more information on adding and managing users, see [User Management and Authentication](#).

Certificate Revocation List Management

Certificate Revocation Lists (CRLs) are a part of the X.509 system that publish lists of certificates that should 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. In pfSense, however, multiple CRLs may be created 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 will show all Certificate Authorities and an option to add a CRL. 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.

Note: CRLs generated using pfSense 2.2.4-RELEASE and later properly include the authorityKeyIdentifier attribute to allow proper functionality with strongSwan for use with IPsec.

Create a new Certificate Revocation List

To create a new CRL:

- Navigate to **System > Cert Manager**, on the **Certificate Revocation** tab.
- Find the row with the CA that the CRL will be created for.
- Click  **Add or Import CRL** at the end of the row to create a new CRL.
- Choose *Create an Internal Certificate Revocation List* for the **Method**.
- Enter a **Descriptive Name** for the CRL, which 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.
- Select the proper CA from the **Certificate Authority** drop-down menu.

- Enter the number of days for which the CRL should be valid in the **Lifetime** box. The default value is 9999 days, or almost 27 and a half years.
- Click **Save**

The browser will be return to the CRL list, and the new entry will be shown there.

Import an Existing Certificate Revocation List

To import a CRL from an external source:

- Navigate to **System > Cert Manager**, on the **Certificate Revocation** tab
- Find the row with the CA that the CRL will be imported for.
- Click  **Add or Import CRL** at the end of the row to create a new CRL.
- Choose *Import an Existing Certificate Revocation List* for the **Method**.
- Enter a **Descriptive Name** for the CRL, which 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.
- Select the proper CA from the **Certificate Authority** drop-down menu.
- 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.

Export a Certificate Revocation List

From the list of CRLs at **System > Cert Manager** on the **Certificate Revocation** tab, a CRL may also be exported.

To export the CRL, click the  icon. The file will download with the descriptive name of the CRL as the file name, and the extension *.crl*.

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.

Revoke a Certificate

A CRL isn't very useful unless it contains revoked certificates. A certificate is revoked by adding the certificate to 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. A screen will be presented that lists any currently revoked certificates, and a control to add new ones.
- Select the certificate from the **Choose a Certificate to Revoke** list.
- 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.
- Click **Add** and the certificate will be added to the CRL.

Certificates can be removed from the CRL using this screen as well:

- 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 adding or 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.

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.
- Erase the pasted content in the **CRL Data** box and replace it with the contents of the new CRL
- 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.

Basic Introduction to X.509 Public Key Infrastructure

One authentication option for VPNs is to use X.509 keys. An in depth discussion of X.509 and Public Key Infrastructure (PKI) is outside the scope of this book, and is the topic of a number of entire books for those interested in details. This chapter provides the very basic understanding necessary for creating and managing certificates in pfSense.

With PKI, first a Certificate Authority (CA) is created. This CA then signs all of the individual certificates in the PKI. The certificate of the CA is used on VPN servers and clients to verify the authenticity of server and client certificates used. 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.

A certificate is considered valid if it has been trusted by a given CA. In this case of VPNs, 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 a unique CA for each VPN should be used.

Certificate Revocation Lists (CRLs) are lists of certificates that have been compromised or otherwise need to be 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.

BACKUP AND RECOVERY

Making Backups in the WebGUI

Making a backup in the WebGUI is simple.

- Navigate to **Diagnostics > Backup & Restore**
- Set the **Backup Area** to *ALL* (the default choice)
- Set any other desired options, such as **Skip RRD** and **Encryption**
- Click **Download Configuration as XML** (Figure *WebGUI Backup*).

Backup & Restore	Config History
Backup Configuration	
Backup area	All
Skip packages	☐ Do not backup package information.
Skip RRD data	☒ Do not backup RRD data (NOTE: RRD Data can consume 4+ megabytes of config.xml space!)
Encryption	☐ Encrypt this configuration file.
Download configuration as XML	

Fig. 11.1: WebGUI Backup

The web browser will then prompt to save the file somewhere on the PC being used to view the WebGUI. It will be named `config-<hostname>-<timestamp>.xml`, but that may be changed before saving the file.

Using the AutoConfigBackup Package

pfSense Gold Subscription users have access to the Automatic Configuration Backup Service via the **AutoConfigBackup** package. The most up to date information on AutoConfigBackup can be found on the pfSense documentation page for the [AutoConfigBackup](#) package.

Functionality and Benefits

When a firewall configuration change is made, it is automatically encrypted with the passphrase entered in the package configuration and uploaded over HTTPS to the AutoConfigBackup servers. Only encrypted configurations are retained on the AutoConfigBackup servers. This gives instant, secure off-site backup of firewall configuration files with no user intervention once the package is configured.

pfSense Version Compatibility

The **AutoConfigBackup** package works with all supported versions of pfSense, and many older versions as well.

Installation and Configuration

To install the package:

- Navigate to **System > Package Manager, Available Packages** tab
- Locate **AutoConfigBackup** in the list
- Click **Install** at the end of the AutoConfigBackup entry
- Click **Confirm** to confirm the installation

The firewall will then download and install the package. Once installed, the package may be found in the menu under **Diagnostics > AutoConfigBackup**

Setting the hostname

Make sure to configure a unique hostname and domain on **System > General Setup**. The configuration entries in AutoConfigBackup are stored by FQDN (Fully Qualified Domain Name, i.e. hostname + domain), so each firewall being backed up must have a unique FQDN, otherwise the system cannot differentiate between multiple installations.

Configuring AutoConfigBackup

The package is configured under **Diagnostics > AutoConfigBackup**. On the **Settings** tab, fill in the settings as follows:

Subscription Username The username for the [pfSense Gold Subscription](#) account

Subscription Password/Confirm The password for the [pfSense Gold Subscription](#) account

Encryption Password/Confirm An arbitrary passphrase used to encrypt the configuration before uploading. This should be a long, complex password to ensure the security of the configuration. The AutoConfigBackup servers only hold encrypted copies, which are useless without this **Encryption Password**

Warning: It is important that the **Encryption Password** be remembered or stored securely outside of the firewall. Without the **Encryption Password**, the configuration file cannot be recovered and the **Encryption Password** is not stored on the server outside of the configuration file.

Testing Backup Functionality

Make a change to force a configuration backup, such as editing and saving a firewall or NAT rule, then click **Apply Changes**. Visit **Diagnostics > AutoConfigBackup, Restore** tab, which will list available backups along with the page that made the change (where available).

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 **Backup Reason**
- Click **Backup**

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.

By default, the package **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, the user is prompted if they want to reboot. If the restored configuration changes anything other than NAT and firewall rules, choose **Yes** and allow the firewall to reboot.

Bare Metal Restoration

If the disk in the firewall fails, as of now the following procedure is required to recover on a new installation.

- Replace the failed disk
- Install pfSense on the new disk
- Configure LAN and WAN, and assign the hostname and domain exactly the same as previously configured
- Install the **AutoConfigBackup** package
- Configure the AutoConfigBackup package as described above, using the same portal account and the same **Encryption Password** used previously.
- Visit the **Restore** tab
- Choose the configuration to restore
- When prompted to reboot after the restoration, do so

Once the firewall has been rebooted, it will be running with the configuration backed up before the failure.

Checking the AutoConfigBackup Status

The status of an AutoConfigBackup run can 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.

Alternate Remote Backup Techniques

The following techniques may also be used to perform backups remotely, but each method has its own security issues which may rule out their use in many places. For starters, these techniques do not encrypt the configuration, which may contain sensitive information. This can result in the 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. The **AutoConfigBackup** package, available with a [pfSense Gold Subscription](#), is a much easier and more secure means of automating remote backups.

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`.

On pfSense 2.2.6 and later, 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 command would be 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:

```
$ 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
```

- Now the script is logged in and can take action. 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 "Submit=download&donotbackuprrd=yes&__csrf_magic=$(head -n 1 csrf2.txt)" \
  https://192.168.1.1/diag_backup.php -O config-hostname-`date +%Y%m%d%H%M%S`.xml
```

Replace the username and password with the credentials for the firewall, and the IP address would be whichever IP address is reachable from the system 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 WebGUI, so adjust the firewall rules accordingly. Performing this over the WAN is not recommended. At a minimum, use HTTPS and restrict access to the WebGUI to a trusted set of public IP addresses. It is preferable to do this locally or over a VPN.

Push with SCP

The configuration file can also be pushed from the pfSense firewall to another UNIX system with `scp`. Using `scp` to push a one-time backup by hand can be useful, but using it in an automated fashion carries some 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
```

In order to push the configuration in an automated manner, 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. If this is done, 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 pfSense system to invoke `scp`. For more details visit the [pfSense Documentation Wiki](#) or search on the forums.

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 pfSense firewall configuration. Automating this method using cron is also possible, but this method requires an SSH key without a 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.)

Restoring from Backups

Backups are not useful without a means to restore them, and by extension, test them. pfSense offers several means for restoring configurations. Some are more involved than others, but each will have the same end result: a running system identical to when the backup was made.

Restoring with the WebGUI

The easiest way for most users to restore a configuration is by using the WebGUI:

- Navigate to **Diagnostics > Backup & Restore**
- Locate the **Restore configuration** section (Figure [WebGUI Restore](#)).
- Select the area to restore (typically *ALL*)

- Click Browse
- Locate the backup file on the local PC
- Click **Restore Configuration**

The configuration will be applied, and the firewall will reboot with the settings obtained from the backup file.

Restore Backup	
Open a pfSense configuration XML file and click the button below to restore the configuration.	
Restore area	All
Configuration file	Browse... No file selected.
Encryption	☐ Configuration file is encrypted.
Restore Configuration	
The firewall will reboot after restoring the configuration.	

Fig. 11.2: WebGUI Restore

While easy to work with, this method does have some prerequisites when dealing with a full restore to a new system. 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 pfSense firewall being restored.

Restoring from the Config History

For minor problems, using one of the internal backups on the pfSense firewall is the easiest way to back out a change. On full installations, the previous 30 configurations are stored in the **Configuration History**, along with the current running configuration. On NanoBSD, 5 configurations are stored. Each row shows the date that 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 will 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

The configuration will be restored, but a reboot is not automatic where required. 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, editing and saving the VPN instance would be enough. The necessary actions to take depend on what changed in the config, but the best way ensure that the full configuration is active would be a reboot. If necessary, reboot the firewall with the new configuration by going to **Diagnostics > Reboot System** and click **Yes**.

Backup & Restore **Config History**

Saved Configurations						
		Date	Version	Size	Configuration Change	Actions
☐		3/14/16 13:53:38	14.7	393 KiB	admin@198.51.100.108: /firewall_aliases_edit.php made unknown change	Current configuration
☐	☐	3/14/16 13:53:28	14.7	393 KiB	admin@198.51.100.108: /vpn_ipsec_settings.php made unknown change	
☐	☐	3/14/16 13:53:20	14.7	393 KiB	admin@198.51.100.108: /vpn_openvpn_server.php made unknown change	
☐	☐	3/14/16 13:34:56	14.7	393 KiB	(system): Installed Service Watchdog package.	

Fig. 11.3: Configuration History

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 .

Config History 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 **Saved Configurations** bar to expand the settings.
- Enter the new number of configurations to retain
- Click **Save**

Along with the configuration count, the amount of space consumed by the current backups is also displayed.

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 allow them to be restored. This is useful if a recent change has locked administrators out of the GUI or taken the system off the network.

Restoring by Mounting the Disk

This method is popular with embedded users. When the CF or disk from the pfSense firewall is attached to a computer running FreeBSD, the drive may be mounted and a new configuration may be copied directly onto the installed system, or a config from a failed system may be copied off.

Note: This can also be performed on a separate pfSense firewall 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/` for both NanoBSD and full installs, but the difference is in the location where this directory resides. For NanoBSD installs, this is on a separate slice, such as `ad0s3` if the drive is `ad0`. Thanks to GEOM (modular storage framework) labels on recent versions of FreeBSD and in use on NanoBSD-based embedded filesystems, this slice may also be accessed regardless of the device name by using the label `/dev/ufs/cf`. For full installs, it is part of the root slice (typically `ad0s1a`). The drive names will vary depending on type and position in the system.

NanoBSD Example

First, connect the CF to a USB card reader on a FreeBSD system or another inactive pfSense system (see the note in the previous section). For most, it will show up as `da0`. Console messages will also be printed reflecting the device name, and the newly available GEOM labels.

Now mount the config partition:

```
# mount -t ufs /dev/da0s3 /mnt
```

If for some reason the GEOM labels are not usable, use the device directly such as `/dev/da0s3`.

Now, copy a config onto the card:

```
# cp /usr/backups/pfSense/config-alix.example.com-20090606185703.xml \
    /mnt/conf/config.xml
```

Then be sure to unmount the config partition:

```
# umount /mnt
```

Unplug the card, reinsert it into the firewall, and turn it on again. The firewall will now be running with the previous configuration.

To copy the configuration from the card, the process is the same but the arguments to the `cp` command are reversed.

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.

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.

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.

Caveats and Gotchas

While the configuration XML file kept by pfSense 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 in some cases. Some passwords must be available in plain text during run time, making secure hashing of those passwords impossible. Any obfuscation would be trivial to reverse for anyone with access to the source code i.e. everyone. A conscious design decision was made in m0n0wall, and continued in pfSense, to leave those passwords in clear to make it exceedingly clear that the file contains sensitive content and must be protected as such. 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 WebGUI 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.

Thanks to the XML-based configuration file used by pfSense, backups are a breeze. All of the settings for the system are held in one single file (see [pfSense 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.

Backup Strategies

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 be helpful, regardless of changes, especially in cases where a manual backup may be missed for one reason or another.

pfSense makes an internal backup upon each change, and we recommend downloading 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. If a [pfSense Gold Subscription](#) is available, backups may be handled easily and automatically using the **AutoConfigBackup** package.

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 book, 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. We also strongly recommend periodically testing 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.

FIREWALL

Firewalling Fundamentals

This section deals primarily with introductory firewall concepts and lays the ground work for understanding how to configure firewall rules using pfSense.

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**

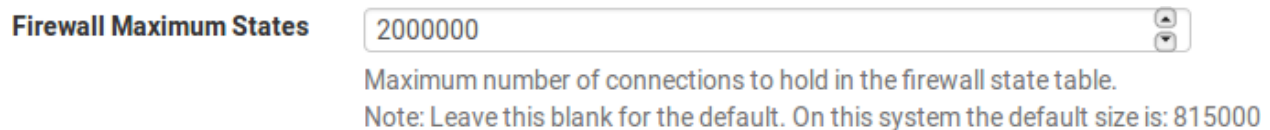


Fig. 12.1: Increased State Table Size to 2,000,000

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.

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 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 12.1: 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.

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.)

Floating

WAN

LAN

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	0/0 B	*		RFC 1918 networks	*	*	*		Block private networks	
	0/0 B	*		Reserved Not assigned by IANA	*	*	*		Block bogon networks	
No rules are currently defined for this interface										
All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule.										

Fig. 12.2: Default WAN Rules

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 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*.

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.

Floating WAN LAN											
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☒	0/0 B	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
☐	0/0 B	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	
☐	0/0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	

Fig. 12.3: Default LAN Rules

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 utilized 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. 12.4: 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.

Rule Information	
Created	7/13/16 12:42:40 by jimp@203.0.113.103
Updated	7/13/16 12:49:33 by admin@198.51.100.6

Fig. 12.5: Firewall Rule Time Stamps

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: Do not confuse Aliases in this context with interface IP aliases, which are a means of adding additional IP addresses to a network interface.

Alias Basics

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.

The following types of aliases can be created:

Host Aliases containing single IP addresses or hostnames

Network Aliases containing CIDR-masked lists of networks, 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 The alias is built from the file at the specified URL but is read only a single time, and then becomes a normal network or port type alias.

URL Table The alias is built from the file at the specified URL but is updated by fetching the list from the URL periodically.

Each alias type is described in more detail throughout this section.

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. URL Table aliases cannot be nested.

Using Hostnames in Aliases

Hostnames can also be used in aliases. Any hostname can be entered into a host or network alias and it will be periodically resolved and updated by the firewall. If a hostname returns multiple IP addresses, all of the returned IP

addresses are added to the alias. This is useful for tracking dynamic DNS entries to allow specific users into services from dynamic IP addresses.

Note: This feature is *not* useful for allowing or disallowing users to large public web sites. Large and busy sites tend to have constantly rotating or random responses to DNS queries so the contents of the alias do not necessarily match up with the response a user will receive when they attempt to 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.

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 200,000. 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 1K per entry to be conservative.

Configuring Aliases

To add an alias:

- Navigate to **Firewall > Aliases**
- Click  **Add**
- Enter a **Name** for the alias. The name may only consist of the characters a-z, A-Z, 0-9 and _.
- Enter a **Description** for the alias itself
- Select the **Type** for the alias. The various types are discussed throughout this section.
- 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.

Host Aliases

Host type aliases contain groups of IP addresses. Figure *Example Hosts Alias* shows an example of a host type alias used to contain a list of public web servers.

Properties			
Name	WebServers The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".		
Description	Public Web Servers A description may be entered here for administrative reference (not parsed).		
Type	Host(s)		
Host(s)			
Hint	Enter as many hosts as desired. Hosts must be specified by their IP address or fully qualified domain name (FQDN). FQDN hostnames are periodically re-resolved and updated. If multiple IPs are returned by a DNS query, all are used. An IP range such as 192.168.1.1-192.168.1.10 or a small subnet such as 192.168.1.16/28 may also be entered and a list of individual IP addresses will be generated.		
IP or FQDN	10.3.1.10	www1	Delete
	10.3.1.11	www2	Delete
	10.3.1.12	www3	Delete
	10.3.1.13	www4	Delete

Fig. 12.6: Example Hosts Alias

Other host type aliases can be nested inside this entry. Hostnames may also be used as entries, as explained previously.

Network Aliases

Network type aliases contain groups of networks or IP address ranges. Single hosts can also be included in network aliases by selecting a /32 network mask for IPv4 addresses or a /128 prefix length for IPv6 addresses. Figure *Example Network Alias* shows an example of a network alias that is used later in this chapter.

Properties			
Name	RemoteAdmin The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".		
Description	Hosts allowed to remotely administrate the firewall A description may be entered here for administrative reference (not parsed).		
Type	Network(s)		
Network(s)			
Hint	Networks are specified in CIDR format. 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. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.		
Network or FQDN	192.168.0.0 / 16	Private management net	Delete
	198.51.100.0 / 24	Data Center	Delete

Fig. 12.7: 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, nothing more, nothing less.

The screenshot shows a single configuration entry for an IP range. The 'Network or FQDN' field contains '10.3.0.100-10.3.0.200'. To its right is a dropdown menu showing '32'. Further right is a 'Description' text field. On the far right is an orange 'Delete' button with a trash icon.

Fig. 12.8: Example IP Range Before

The screenshot shows a table of expanded IP ranges. Each row contains a 'Network or FQDN' field with a CIDR address, a dropdown menu with a value, a 'Description' field, and a 'Delete' button. The entries are as follows:

Network or FQDN	Value	Description	Action
10.3.0.100	30	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete
10.3.0.104	29	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete
10.3.0.112	28	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete
10.3.0.128	26	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete
10.3.0.192	29	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete
10.3.0.200	32	Entry added Wed, 13 Jul 2016 16:18:40 -0400	Delete

Fig. 12.9: Example IP Range After

Port Aliases

Port type aliases contain groups of ports and port ranges. 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.

The screenshot shows the configuration form for a port alias. It is divided into two main sections: 'Properties' and 'Port(s)'. The 'Properties' section includes fields for 'Name' (WebPorts), 'Description' (Ports used by web servers), and 'Type' (Port(s)). The 'Port(s)' section includes a 'Hint' and a table of port entries.

Properties

- Name:** WebPorts. The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
- Description:** Ports used by web servers. A description may be entered here for administrative reference (not parsed).
- Type:** Port(s)

Port(s)

Hint: Enter ports as desired, with a single port or port range per entry. Port ranges can be expressed by separating with a colon.

Port	Protocol	Action
80	HTTP	Delete
443	HTTPS	Delete

Fig. 12.10: 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, a URL is set which points to a text file that contains 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.

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 file 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.

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, a list of matching aliases is displayed. 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. 12.11: 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. 12.12: 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.

Rules (Drag to Change Order)										
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐	✓ 0/0 B	IPv4 TCP	*	*	WebServers	WebPorts	*	none		Allow access to WebPorts on WebServers

Fig. 12.13: 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.

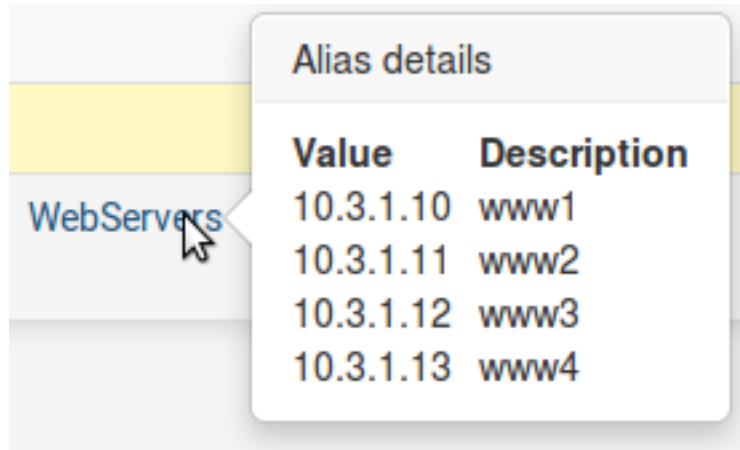


Fig. 12.14: Hovering Shows Hosts Contents

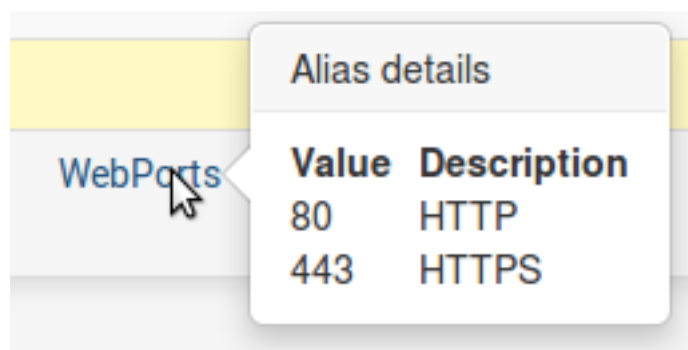


Fig. 12.15: Hovering Shows Ports Contents

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. 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. 12.16: 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, 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*).

Properties			
Name	RemoteAdminPorts The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".		
Description	Ports used for firewall management A description may be entered here for administrative reference (not parsed).		
Type	Port(s)		

Port(s)			
Hint	Enter ports as desired, with a single port or port range per entry. Port ranges can be expressed by separating with a colon.		
Port	443	WebGUI (HTTPS)	Delete
	22	SSH	Delete

Fig. 12.17: 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*).

The resulting aliases are shown in Figure *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

Properties	
Name	RemoteAdmin The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	Hosts allowed to remotely administrate the firewall A description may be entered here for administrative reference (not parsed).
Type	Network(s)
Network(s)	
Hint	Networks are specified in CIDR format. 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. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.
Network or FQDN	192.168.0.0 / 16 Private management net Delete 198.51.100.0 / 24 Data Center Delete

Fig. 12.18: Alias For Management Hosts

RemoteAdmin	192.168.0.0/16, 198.51.100.0/24	Hosts allowed to remote admin
RemoteAdminPorts	443, 22	Ports used for firewall management

Fig. 12.19: Alias List

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. 12.20: Example Restricted Management LAN Rules

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.

	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. 12.21: Restricted Management LAN Rules Alternate Example

Anti-lockout  Disable webConfigurator anti-lockout rule

When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

Fig. 12.22: 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 might indicate either spoofed traffic or an unused subnet that has been hijacked for malicious use. pfSense provides two bogons lists that are updated as needed, one for IPv4 bogon networks and one for IPv6 bogon networks. If **Block bogon networks** is enabled, the firewall will fetch an updated bogons list on the first day of each month from `files.pfsense.org`. 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 very frequently, and new IP address assignments are removed from the bogons list months before they are actually used, so a monthly update is adequate. If the list must be updated 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.

Make sure the firewall can resolve DNS host names, otherwise the update will fail. To ensure the firewall resolve DNS, browse to **Diagnostics > DNS Lookup**, and try to resolve `files.pfsense.org`. If that works, then go to **Diagnostics > Test Port** and try to connect to `files.pfsense.org` on port 80 as demonstrated in Figure *figure-testing-connectivity-for-bogon-updates*.

Port test to host: files.pfsense.org Port: 80 successful.

Test Port

Hostname

Port

Source Port

Remote text ☐ Show remote text
Shows the text given by the server when connecting to the port. If checked it will take 10+ seconds to display in a panel below this form.

Source Address
Select source address for the trace.

IP Protocol
If IPv4 or IPv6 is forced and a hostname is used that does not contain a result using that protocol, it will result in an error. For example if IPv4 is forced and a hostname is used that only returns an AAAA IPv6 IP address, it will not work.

 Test

Fig. 12.23: Testing Connectivity for Bogon Updates

Forcing a bogons 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 firewall web interface at **Diagnostics > Tables**, by selecting *bogons* or *bogonsv6* then click  **Update**.

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 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.

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 book 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.

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. 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.

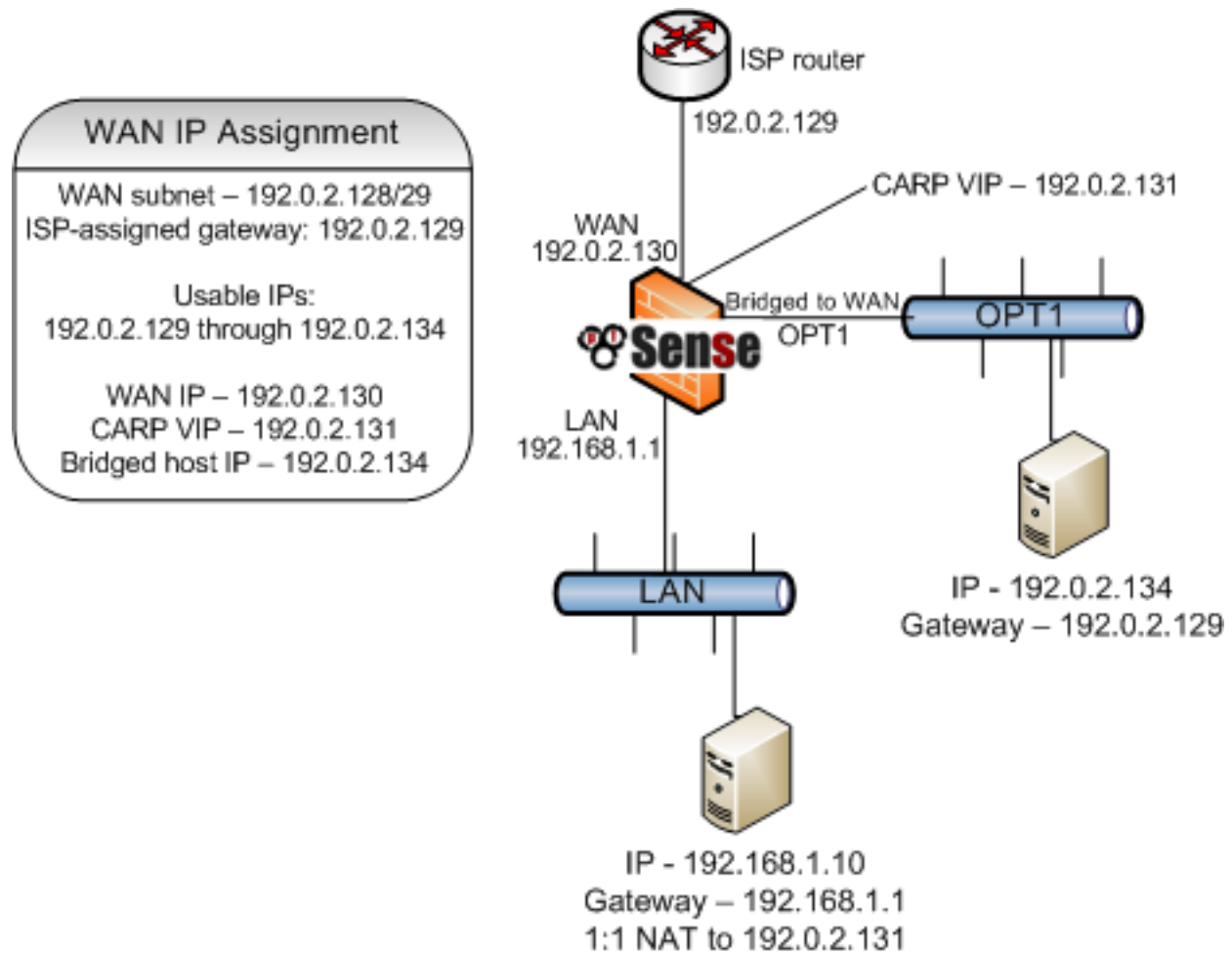


Fig. 12.24: Multiple Public IP addresses In Use Single IP Subnet

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](#).

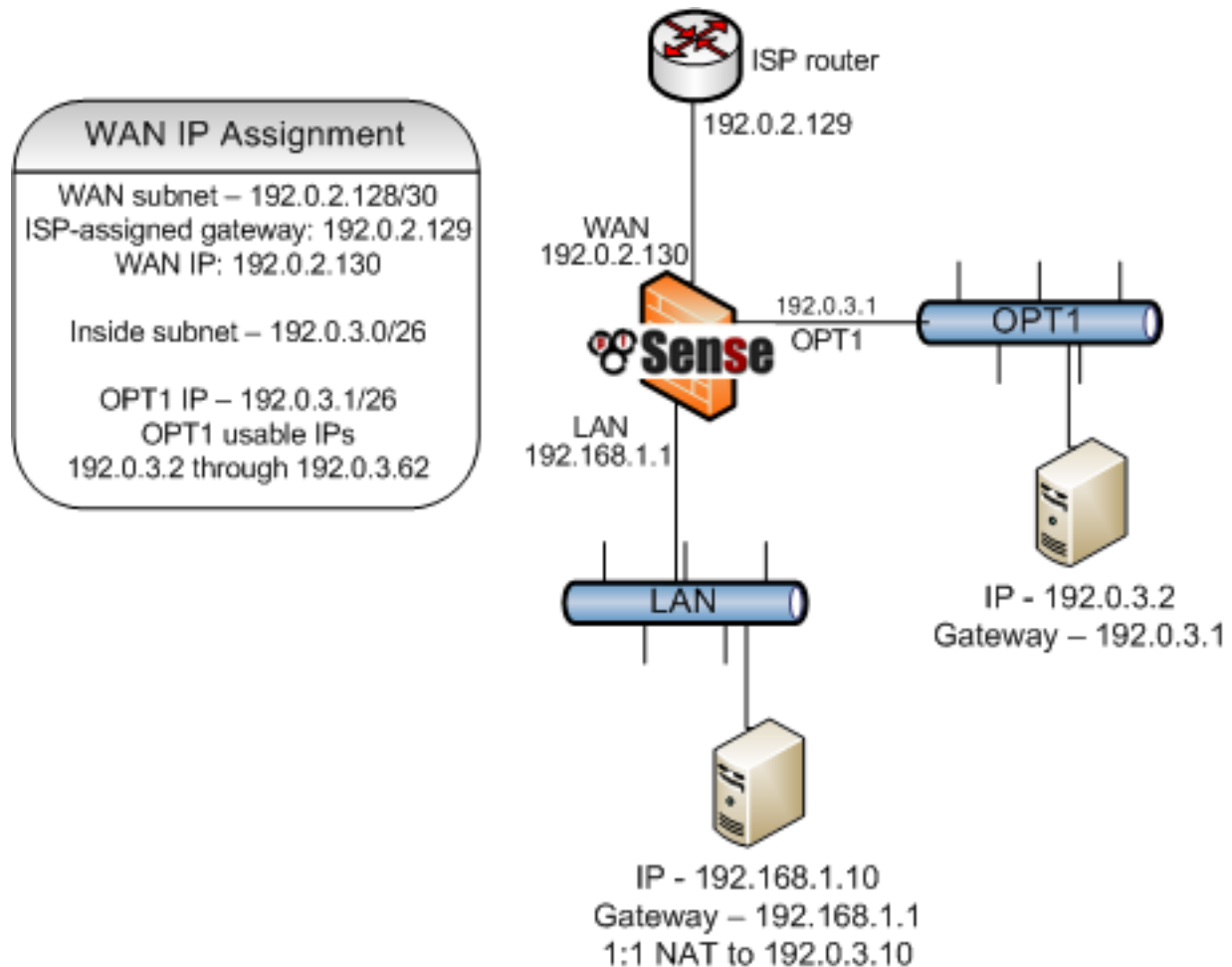


Fig. 12.25: Multiple Public IP Addresses Using Two IP Subnets

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](#).

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 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 be 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.

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.

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 [Interface](#).

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.

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.

Schedule Information

Schedule Name

BusinessHours

Description

Normal Business Hours

Month

August_16

Date

August_2016

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Click individual date to select that date only. Click the appropriate weekday Header to select all occurrences of that weekday.

Time

9

00

17

00

Time range description

Work Week

+ Add Time

Clear selection

Fig. 12.26: Adding a Time Range

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*.

Configured Ranges

Mon - Fri	9:00	17:00	Work Week	Delete
Day(s)	Start time	Stop time	Description	

Fig. 12.27: 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.

Schedules			
Name	Range: Date / Times / Name	Description	Actions
 BusinessHours	Mon - Fri / 9:00-17:00 / Work Week	Normal Business Hours	 

 Indicates that the schedule is currently active.

Fig. 12.28: 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](#).

Schedule BusinessHours

Leave as 'none' to leave the rule enabled all the time.

Fig. 12.29: 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. 12.30: 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 Firewall Logs

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 *Changing Log Settings* for details.

Like other logs in pfSense, the firewall logs only keep a certain number of records using the binary circular log format, *clog*. If the needs of an organization require a permanent record of firewall logs for a longer period of time, see *System Logs* for information on copying these log entries to a syslog server as they happen.

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 *Changing 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)

Time The time that the packet arrived.

Interface Where the packet entered the firewall.

Source The source IP address and port.

Destination The destination IP address and port.

Protocol The protocol of the packet, e.g. ICMP, TCP, UDP, etc.

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. 12.31: Example Log Entries Viewed From The WebGUI

The **Action** icon is a link which will lookup and display the rule that caused the log entry. More often than not, this says “Default Deny Rule”, but when troubleshooting rule issues it can help narrow down suspects.

Tip: On the **Settings** tab under **Status > System Logs**, this rule description can be configured to show in the log entries directly. The firewall can display the description in a separate column or a separate row. See *Changing Log Settings* for details.

Next to the source and destination IP addresses is . When this icon is clicked the firewall will 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.

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. Common flags include:

S – SYN Synchronize sequence numbers. Indicates a new connection attempt when only SYN is set.

A – ACK Indicates ACKnowledgment of data. These are replies to let the sender know data was received OK.

F – FIN Indicates there is no more data from the sender, closing a connection.

R – 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.

See also:

There are several other flags and their meanings are outlined in many materials on the TCP protocol. The [Wikipedia article on TCP](#) has more information.

The log output shown in the GUI may be filtered 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.

Adding Firewall Rules from the Log View (Easy Rule)

Easy Rule makes it simple to add firewall rules quickly from the firewall log view.

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.

Using Easy Rule to add firewall rules from the shell

The shell version of Easy Rule, `easyrule`, can be used to add a firewall rule from a shell prompt. When the `easyrule` command is run without parameters, the a usage message is printed to explain its syntax.

The way it adds a block rule using an alias, or a precise pass rule specifying the protocol, source, and destination, work similar to the GUI version. For example, to add a block rule, run:

```
# easyrule block wan 1.2.3.4
```

A pass rule must be more precise:

```
# easyrule pass wan tcp 1.2.3.4 192.168.0.4 80
```

Viewing from the Console Menu

The raw logs may be viewed and followed in real time from the `filter.log` file using option 10 from the console menu. 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,igbl,match,block,in,4,0x10,,128,0,0,none,17,
```

This shows that rule id 1000000103 was matched, which resulted in a block action on the `igbl` 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:

The [format of the filter log file](#) is described in detail on the pfSense documentation wiki.

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 clog file, the log entries can be quite complex and verbose.

Viewing the current contents of the log file

The filter log is contained in a binary circular log so traditional tools like `cat`, `grep`, and so on cannot be used on the file directly. The log must be read back with the `clog` program, and may then be piped through another program.

To view the entire contents of the log file, run the following command:

```
# clog /var/log/filter.log
```

To restrict the log output to the last few lines, pipe it through `tail`:

```
# clog /var/log/filter.log | tail
```

Following the log output in real time

To “follow” the output of the log file, use the `-f` parameter to `clog`. This is the equivalent of `tail -f` for those used to working with normal log files on UNIX systems:

```
# clog -f /var/log/filter.log
```

This will output the entire contents of the log file but does not quit afterward. It will instead wait for more entries and print them as they happen. This output may also be piped to other commands as needed.

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:

```
# clog /var/log/filter.log | filterparser.php
```

The log entries output one per line, with simplified output:

```
Aug  3 08:59:02 block igbl 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 ID number for an entry is displayed. This rule number can be used to find the rule which caused the match. In the following example, what 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.

Why are there blocked log entries for legitimate connections?

Sometimes log entries are present that, while labeled with the “Default deny” rule, appear as though they belong to legitimate connections. The most common example is seeing a connection blocked involving a web server.

This is likely to happen when a TCP FIN packet, which would normally close the connection, arrives after the connection’s state has been removed or when an ACK is received outside the acceptable window time. This happens because on occasion a packet will be lost or delayed and the retransmits will be blocked because the firewall has already closed the connection.

These log entries are harmless and do not indicate an actual blocked connection. All stateful firewalls do this, though some don’t generate log messages for this blocked traffic even when all blocked traffic is logged.

This behavior will be present on occasion even if “allow all” style rules exist on all of the firewall interfaces because a rule set to “allow all” for TCP connections only allows TCP SYN packets to create a state. All other TCP traffic will either be part of an existing state in the state table, or will be packets with spoofed or otherwise invalid TCP flags.

A special variation of this that *can* indicate trouble is when asymmetric routing exists on a network. In those cases log entries will be present showing TCP : SA (SYN+ACK) packets being blocked rather than FIN or RST. See *Bypass Firewall Rules for Traffic on Same Interface* and *Static Route Filtering* for information on how to handle asymmetric routing.

How Do I Block access to a Web Site?

A question we get asked very often is “How do I block access to a web site?”, or to be more accurate: “How do I block access to Facebook?” And it isn’t always an easy question to answer. There are several possible tactics to accomplish the goal, some are discussed elsewhere in the book.

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.

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. 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 very 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.

The pfBlocker package offers mechanisms which can be useful in this area, such as DNSBL, geographic IP address blocking, and automation of the AS lookup process.

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](#).

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. 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.

Troubleshooting Firewall Rules

This section provides guidance for troubleshooting issues with firewall rules.

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.

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.

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.

Review Rule Ordering

Remember that for interface tab rules, the first matching rule wins – no further rules are evaluated.

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.

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.

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.

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.

Other Causes

There are other pitfalls in firewall rules, NAT, routing, and network design that can interfere with connectivity. See the doc wiki article on [Connectivity Troubleshooting](#) for more suggestions.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the June 2016 hangout on Connectivity Troubleshooting which contains much more detailed troubleshooting procedures.

One of the primary functions performed by pfSense is filtering traffic. This chapter covers fundamentals of firewalling, best practices, and required information necessary to configure firewall rules.

NETWORK ADDRESS TRANSLATION

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” after countless complaints from confused users. Similar functionality is also called “Destination NAT” in other products. However, “Port Forward” a misnomer, as port forward rules can redirect GRE and ESP protocols 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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) 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.

Risks of Port Forwarding

In a default configuration, pfSense does not let in 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 the corresponding firewall rule. It 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.

Port Forwarding and Local Services

Port forwards take precedence over any services running locally on the firewall, such as the web interface, SSH, and so on. 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.

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.

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 begin adding a port forward entry, click  **Add** button to reach the **Port Forward** editing screen. The following options are available for port forwards:

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 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.

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 being forwarded will be *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.

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. 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. An alias here, but the alias must only contain a **single address**. If the alias contains multiple addresses, the port will be forwarded to each host alternately, which is not what most people want. To setup load balancing for one port to multiple internal servers, see [Server Load Balancing](#).

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. This is the best choice for most use cases. If this option is chosen, after the rule is saved a link is placed here which leads to the associated firewall rule.

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, so they do not work effectively with Multi-WAN.

- 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 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.

Disabled

☐ Disable this rule

No RDR (NOT)

☐ Disable redirection for traffic matching this rule

This option is rarely needed. Don't use this without thorough knowledge of the implications.

Interface

WAN

Choose which interface this rule applies to. In most cases "WAN" is specified.

Protocol

TCP

Choose which protocol this rule should match. In most cases "TCP" is specified.

Source

Display Advanced

Destination

☐ Invert match.

WAN address

Type

Address/mask

Destination port range

HTTP

From port

Custom

HTTP

To port

Custom

Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.

Redirect target IP

10.3.0.15

Enter the internal IP address of the server on which to map the ports.
e.g.: 192.168.1.12

Redirect target port

HTTP

Port

Custom

Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically).
This is usually identical to the "From port" above.

Description

HTTP to web server

A description may be entered here for administrative reference (not parsed).

No XMLRPC Sync

☐ Do not automatically sync to other CARP members

This prevents the rule on Master from automatically syncing to other CARP members. This does NOT prevent the rule from being overwritten on Slave.

NAT reflection

Use system default

Filter rule association

Add associated filter rule

The "pass" selection does not work properly with Multi-WAN. It will only work on an interface containing the default gateway.

Fig. 13.1: Port Forward Example

Port Forward1:1OutboundNPT

Rules											
	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description	Actions	
☐	☒ WAN	TCP	*	*	WAN address	80 (HTTP)	10.3.0.15	80 (HTTP)	HTTP to web server	☒	☐

Fig. 13.2: Port Forward List

☐	☒	0/1 KiB	IPv4 TCP	*	*	10.3.0.15	80 (HTTP)	*	none	NAT HTTP to web server	☒	☐	☐	☐

Fig. 13.3: Port Forward Firewall Rule

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.

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.

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. 13.4: 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.

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.

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 actual squid proxy itself. They must be in the correct order in the list of port forwards: The negate rule first, then the redirect.

No RDR (NOT)	☒ Disable redirection for traffic matching this rule This option is rarely needed. Don't use this without thorough knowledge of the implications.		
Interface	LAN Choose which interface this rule applies to. In most cases "WAN" is specified.		
Protocol	TCP Choose which protocol this rule should match. In most cases "TCP" is specified.		
Source	Hide Advanced		
Source	☐ Invert match.	Single host or alias Type	10.3.0.10 / Address/mask
Source port range	Any From port	Custom To port	Custom
Specify the source port or port range for this rule. This is usually random and almost never equal to the destination port range (and should usually be 'any'). The 'to' field may be left empty if only filtering a single port.			
Destination	☐ Invert match.	Any Type	Address/mask
Destination port range	HTTP From port	Custom To port	Custom
Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.			
Description	Do not redirect HTTP from Squid A description may be entered here for administrative reference (not parsed).		

Fig. 13.5: Example Redirect Port Forward (Negation)

No RDR (NOT)	☐ Disable redirection for traffic matching this rule This option is rarely needed. Don't use this without thorough knowledge of the implications.		
Interface	LAN Choose which interface this rule applies to. In most cases "WAN" is specified.		
Protocol	TCP Choose which protocol this rule should match. In most cases "TCP" is specified.		
Source	Display Advanced		
Destination	☐ Invert match.	Any Type	Address/mask
Destination port range	HTTP From port	Custom To port	Custom
Specify the port or port range for the destination of the packet for this mapping. The 'to' field may be left empty if only mapping a single port.			
Redirect target IP	10.3.0.10 Enter the internal IP address of the server on which to map the ports. e.g.: 192.168.1.12		
Redirect target port	Other Port	3128 Custom	Specify the port on the machine with the IP address entered above. In case of a port range, specify the beginning port of the range (the end port will be calculated automatically). This is usually identical to the "From port" above.
Description	Redirect HTTP to Squid A description may be entered here for administrative reference (not parsed).		

Fig. 13.6: Example Redirect Port Forward

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.

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.

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.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 13.7: 1:1 NAT Entry.

Edit NAT 1:1 Entry			
Disabled	☐ Disable this rule When disabled, the rule will not have any effect.		
No BINAT (NOT)	☐ Do not perform binat for the specified address Excludes the address from a later, more general, rule.		
Interface	WAN Choose which interface this rule applies to. In most cases "WAN" is specified.		
External subnet IP	198.51.100.210 Enter the external (usually on a WAN) subnet's starting address for the 1:1 mapping. The subnet mask from the internal address below will be applied to this IP address.		
Internal IP	☐ Not Invert the sense of the match.	Single host Type	10.3.1.15 / 31 Address/mask
Enter the internal (LAN) subnet for the 1:1 mapping. The subnet size specified for the internal subnet will be applied to the external subnet.			
Destination	☐ Not Invert the sense of the match.	Any Type	Address/mask
The 1:1 mapping will only be used for connections to or from the specified destination. Hint: this is usually "Any".			
Description	Mail Server A description may be entered here for administrative reference (not parsed).		
NAT reflection	Use system default		

Fig. 13.7: 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 13.1: /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 13.2: /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*.

Edit NAT 1:1 Entry

Disabled
☐ Disable this rule
When disabled, the rule will not have any effect.

No BINAT (NOT)
☐ Do not perform binat for the specified address
Excludes the address from a later, more general, rule.

Interface
WAN
Choose which interface this rule applies to. In most cases "WAN" is specified.

External subnet IP
198.51.100.64
Enter the external (usually on a WAN) subnet's starting address for the 1:1 mapping. The subnet mask from the internal address below will be applied to this IP address.

Internal IP
☐ Not
Invert the sense of the match.

Network
10.3.1.64 / 30
TypeAddress/mask

Enter the internal (LAN) subnet for the 1:1 mapping. The subnet size specified for the internal subnet will be applied to the external subnet.

Destination
☐ Not
Invert the sense of the match.

Any
TypeAddress/mask

The 1:1 mapping will only be used for connections to or from the specified destination. Hint: this is usually "Any".

Description
.64 through .67 range
A description may be entered here for administrative reference (not parsed).

NAT reflection
Use system default

Fig. 13.8: 1:1 NAT entry for /30 CIDR range

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, 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.

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 book 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.

Traffic from LAN to WAN is processed as described in the following more detailed list. 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, then rules for interface groups including the LAN interface, then 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, then rules for interface groups including the WAN interface, then 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.

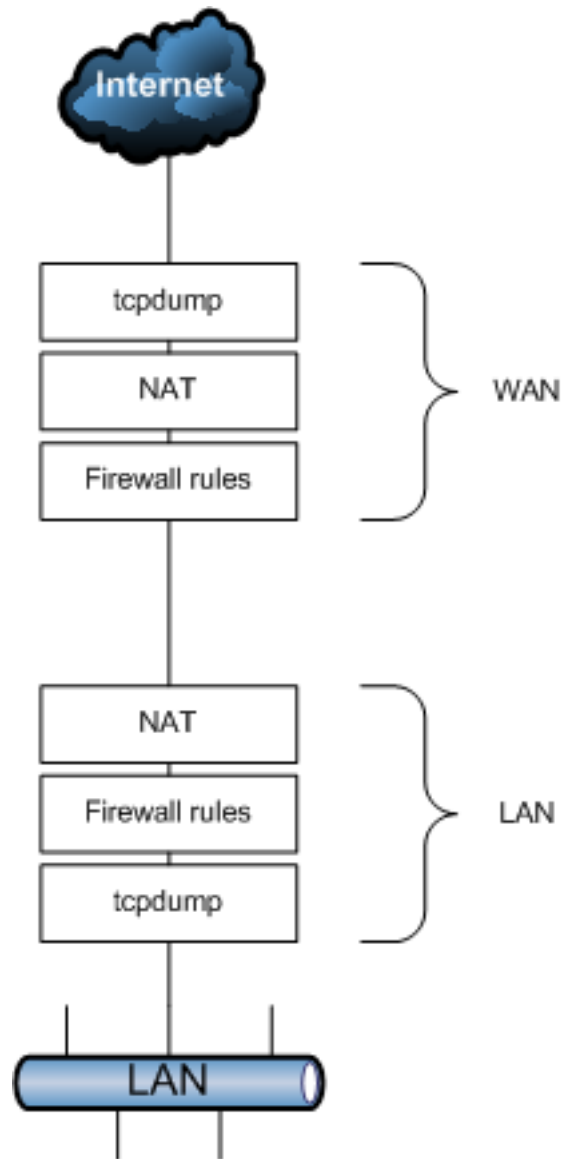


Fig. 13.9: Ordering of NAT and Firewall Processing

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. 13.10: 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.

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. Many commercial and open source firewalls do not support this functionality at all. When possible, split DNS is the preferred means of accessing resources so that the firewall is not involved in accessing internal services internally. pfSense has good support for NAT reflection, though some environments will require a split DNS infrastructure to accommodate this functionality. Split DNS is covered at the end of this section in *Split DNS*.

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.

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  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**

Host Override Options	
Host	 Name of the host, without the domain part e.g.: "myhost"
Domain	example.com Domain of the host e.g.: "example.com"
IP Address	10.3.0.20 IP address of the host e.g.: 192.168.100.100 or fd00:abcd::1
Description	override for example.com web site A description may be entered here for administrative reference (not parsed).

Additional Names for this Host			
www	example.com	Alias www.example.com	 Delete
Host name	Domain	Description	

Fig. 13.11: 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.

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 through which traffic will exit the firewall. That is handled by the routing table ([Static Routes](#)) or policy routing ([Policy routing](#)).

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.

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.

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.

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.

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.

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.

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.

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.

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, and how to work around these issues where possible.

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.

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.

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.

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.

IPv6 Network Prefix Translation (NPt)

Network Prefix Translation, or NPt for short, works similarly to 1:1 NAT but operates on IPv6 addresses 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.

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 [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.

Edit NAT NPt Entry	
Disabled	☐ Disable this rule
Interface	HENETV6DSL Choose which interface this rule applies to. Hint: Typically the "WAN" is used here.
Internal IPv6 prefix	☐ Not Use this option to invert the sense of the match.
Address	2001:db8:1111:2222:: / 64 Internal (LAN) ULA IPv6 Prefix for the Network Prefix translation. The prefix size specified for the internal IPv6 prefix will be applied to the external prefix.
Destination IPv6 prefix	☐ Not Use this option to invert the sense of the match.
Address	2001:db8:3333:4444:: / 64 Global Unicast routable IPv6 prefix
Description	Translate LAN to IPv6 DSL/WAN2 A description may be entered here for administrative reference (not parsed).

Fig. 13.12: NPt Example

Troubleshooting

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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) 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.

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 Logs](#)). 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.

Edit Advanced Outbound NAT Entry

Disabled ☒ Disable this rule

Do not NAT ☐ Enabling this option will disable NAT for traffic matching this rule and stop processing Outbound NAT rules
In most cases this option is not required.

Interface LAN
Choose which interface this rule applies to. In most cases "WAN" is specified.

Protocol TCP
Choose which protocol this rule should match. In most cases "any" is specified.

Source Any 10.3.0.0 / 24
Type Source network for the outbound NAT mapping. Port

Destination Network 10.3.0.6 / 24 80
Type Destination network for the outbound NAT mapping. Port

☐ Not
Invert the sense of the destination match.

Translation

Address Interface Address

Port ☐ Static port
Enter the source port or range for the outbound NAT mapping.

Fig. 13.13: 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 to hosts behind the portal only work when the target system is logged 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.

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.

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.

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 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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the May 2016 Hangout on NAT with pfSense 2.3 and the earlier August 2014 Hangout on Network Address Translation.

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.

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.

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.

ROUTING

Gateways

Gateways are the key to routing; They are systems 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 system will connect 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 normal networks, gateways always reside in the same subnet as one of the interfaces on a system. 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.

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.

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, but a dynamic gateway may not be deleted.

To add or manage gateways:

- Navigate to **System > Routing**
- Click the **Gateways** tab
- Click  **Add** at the top or bottom of the list to create a new gateway

- Click  next to an entry to edit an existing gateway
- Click  next to an entry to delete a gateway
- Click  to disable an active gateway
- Click  to enable a disabled gateway

The individual options for gateways are discussed in detail in the next section.

Gateway Settings

When adding or editing a gateway, a screen is presented that lists all of the options for controlling gateway behavior. The only required settings are the **Interface**, the **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**.

Default Gateway

When selected, this gateway is treated as the default gateway for the system. The default gateway is the gateway of last resort. It is used when there are no other more specific routes. The firewall can have one IPv4 default gateway and one IPv6 default gateway.

Disable Gateway Monitoring

By default, the system will ping each gateway once per second 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.

Monitor IP

The **Monitor IP** address option configures the IP address used to determine the gateway status. By default the system 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 DSL CPE. In cases such as that 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 returning 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.

Advanced

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, if two WANs have different amounts of bandwidth, the **Weight** parameter adjusts the ration at which the WANs are used. For example if WAN1 has 5Mbit/s and WAN2 has 10Mbit/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 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. The default values are **From** 300 and **To** 500.

Some other common situations may require adjusting these values. For instance some DSL lines run fine 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 operating acceptably. 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 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. The default values are **From** 10 and **To** 20.

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 is 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.

Gateway Groups

Gateway Groups define sets of gateways to be used for failover or load balancing. Gateway Groups can also be used as **Interface** values in some areas of the GUI for service failover, such as OpenVPN, IPsec, and Dynamic DNS.

For information on setting up those features, see [Multiple WAN Connections](#).

Static Routes

Static routes are used when hosts or networks are reachable through a router other than the default gateway. pfSense knows about the networks directly attached to it, and reaches all other networks as directed by its routing table. In networks where an internal router connects additional internal subnets, a static route must be defined for that network to be reachable. 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 found under **System > Routing** on the **Routes** tab.

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 follows:
 - Destination Network** Specifies the network and subnet mask that is reachable using this route.
 - Gateway** Defines the router through which this network is reached.
 - Disabled** Check if the static route should not be used, only defined.
 - Description** Some text to describe the route, its purpose, etc.
- Click **Save**
- Click **Apply Changes**

To manage existing routes:

- Navigate to **System > Routing** on the **Routes** tab
- Click  next to an entry to edit an existing route
- Click  next to an entry to delete a route
- Click  to disable an active route
- Click  to enable a disabled route
- Click **Apply Changes**

Example Static Route

Figure *Static Routes* illustrates a scenario where a static route is required.



Fig. 14.1: Static Routes

Because the 192.168.2.0/24 network in Figure *Static Routes* is not on an interface directly connected to pfSense, a static route is needed so the firewall knows 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.

Firewall rule adjustments may also be required. If custom LAN rules are used, they must allow traffic to pass from a source of the networks reachable via static routes on LAN.

Edit Route Entry	
Destination network	192.168.2.0 / 24 Destination network for this static route
Gateway	OtherRouter - 192.168.1.254 Choose which gateway this route applies to or add a new one first
Disabled	☐ Disable this static route Set this option to disable this static route without removing it from the list.
Description	 A description may be entered here for administrative reference (not parsed).

Fig. 14.2: Static Route Configuration

Bypass Firewall Rules for Traffic on Same Interface

In many situations when 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.

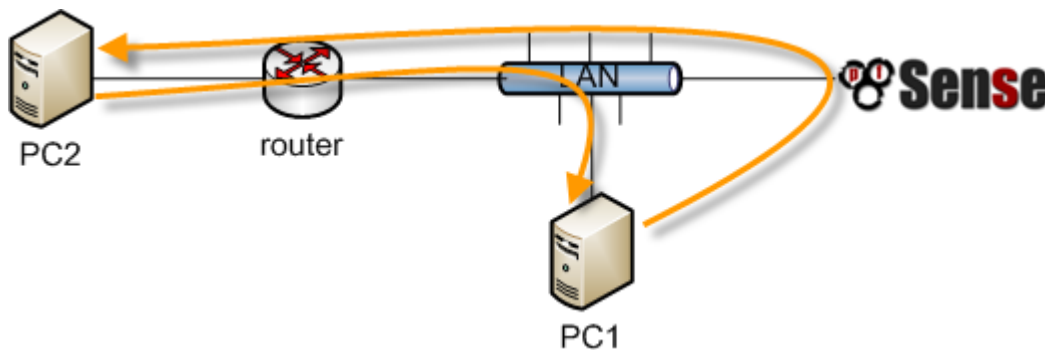


Fig. 14.3: Asymmetric Routing

Traffic from PC1 to PC2 will go through pfSense since it is the default gateway for PC1, but traffic in the opposite direction will go directly from the router to PC1. Since pfSense is a stateful firewall, it must see traffic for the entire connection to be able to filter traffic properly. With asymmetric routing such as 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.

In asymmetric routing scenarios, there is an option that may 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.

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.

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 [Providing Redundancy Without NAT](#).

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 14.1: 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 14.2: 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	

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](#).

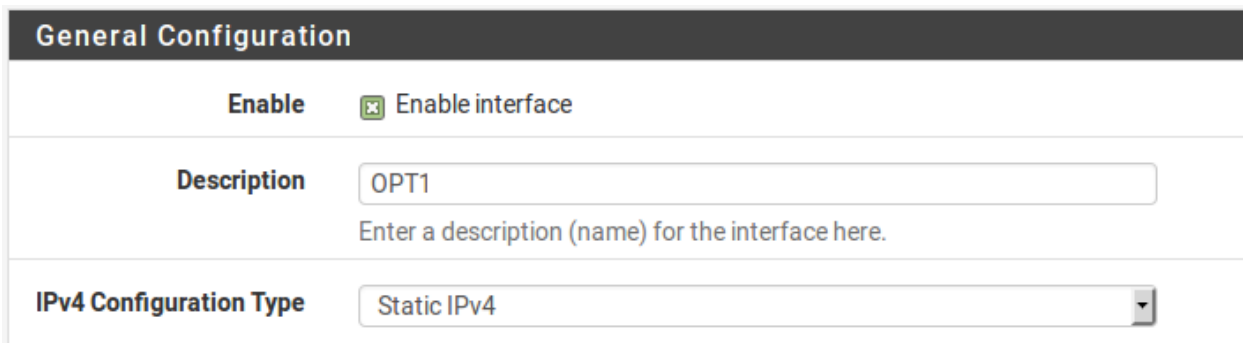


The image shows a 'Static IPv4 Configuration' form. It has two main sections. The first section is for the 'IPv4 Address', with a text input field containing '198.51.100.66', a dropdown menu for the subnet mask set to '30', and a small arrow icon. The second section is for the 'IPv4 Upstream gateway', with a dropdown menu showing 'WANGW - 198.51.100.65' and a green button with a plus sign and the text 'Add a new gateway'.

Fig. 14.4: WAN IP and Gateway Configuration

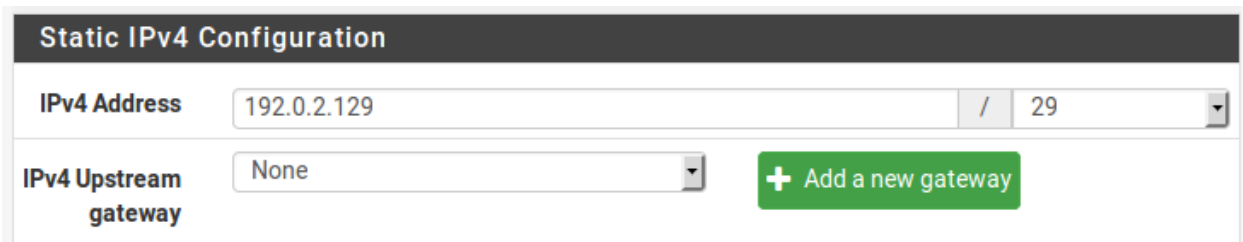
Configure OPT1

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*.



The image shows a 'General Configuration' form for the OPT1 interface. It has three main sections. The first section is for 'Enable', with a checkbox labeled 'Enable interface' that is checked. The second section is for 'Description', with a text input field containing 'OPT1' and a hint text 'Enter a description (name) for the interface here.' The third section is for 'IPv4 Configuration Type', with a dropdown menu set to 'Static IPv4'.

Fig. 14.5: Routing OPT1 Interface Configuration



The image shows a 'Static IPv4 Configuration' form for the Routing OPT1 IP Address. It has two main sections. The first section is for the 'IPv4 Address', with a text input field containing '192.0.2.129', a dropdown menu for the subnet mask set to '29', and a small arrow icon. The second section is for the 'IPv4 Upstream gateway', with a dropdown menu set to 'None' and a green button with a plus sign and the text 'Add a new gateway'.

Fig. 14.6: Routing OPT1 IP Address Configuration

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.

General Logging Options

Mode

☐ Automatic outbound NAT rule generation. (IPsec passthrough included)
 ☒ Hybrid Outbound NAT rule generation. (Automatic Outbound NAT + rules below)
 ☐ Manual Outbound NAT rule generation. (AON - Advanced Outbound NAT)
 ☐ Disable Outbound NAT rule generation. (No Outbound NAT rules)

Save

Mappings

	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description	Actions
☐	☒ WAN	192.0.2.128/29	*	*	*	NO NAT	*	☒	Do not NAT public subnet	  

Automatic Rules:

	Interface	Source	Source Port	Destination	Destination Port	NAT Address	NAT Port	Static Port	Description
☒	WAN	127.0.0.0/8 192.168.1.0/24 192.0.2.128/29	*	*	500	WAN address	*	☒	Auto created rule for ISAKMP
☒	WAN	127.0.0.0/8 192.168.1.0/24 192.0.2.128/29	*	*	*	WAN address	*	☒	Auto created rule

Fig. 14.7: Outbound NAT Configuration

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.

Note: Traffic will flow from LAN to this public subnet by default without NAT. If this behavior is not desired, adjust

Floating	WAN	LAN	OPT1

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐		0/0 B	IPv4 *	*	LAN net	*	*	none		Reject all to LAN	
☐		0/0 B	IPv4 TCP	OPT1 net	*	80 (HTTP)	*	none		Allow HTTP outbound	
☐		0/0 B	IPv4 TCP/UDP	OPT1 net	*	53 (DNS)	*	none		Allow DNS requests to the firewall itself	
☐		0/0 B	IPv4 ICMP echoreq	OPT1 net	*	*	*	none		Allow ICMP echo (ping) to the firewall for diagnosti	

Fig. 14.8: OPT1 Firewall Rules

☐		0/0 B	IPv4 TCP	*	*	192.0.2.130	80 (HTTP)	*	none	Allow HTTP to server1	

Fig. 14.9: WAN Firewall Rules

the LAN firewall and NAT rules accordingly. Additionally, policy routing may need to be bypassed to allow from LAN to this interface.

Routing Protocols

At the time of this writing, three routing protocols are supported with pfSense:

- RIP (Routing Information Protocol)
- BGP (Border Gateway Protocol)
- OSPF (Open Shortest Path First).

This section is light on details, and presumes understanding of the routing protocols as a prerequisite. An in depth discussion of routing protocols is outside the scope of this book.

RIP

RIP is part of the routed package. To install it:

- Navigate to **System > Package Manager**
- Click **Available Packages**
- Locate **routed** in the list, or search for it
- Click the **Install** to the right of the **routed** package entry.
- Click **Confirm**
- Wait for the installation to complete
- Navigate to **Services > RIP**

To configure RIP:

- Check the **Enable RIP** box
- Choose the **Interfaces** RIP will listen and send routing updates on
- Select the **RIP version**
- Enter a **RIPv2 password** if RIPv2 is in use and requires a password on the network.
- Click **Save**

RIP will immediately launch and start sending and receiving routing updates on the specified interfaces.

BGP

A BGP package using [OpenBGPD](#) from OpenBSD is available. To install it:

- Navigate to **System > Package Manager**
- Click **Available Packages**
- Locate **OpenBGPD** in the list, or search for it
- Click the  **Install** to the right of the **OpenBGPD** package entry.
- Click  **Confirm**
- Wait for the installation to complete
- Navigate to **Services > OpenBGPD**

BGP is a complex beast, and describing it in detail is outside the scope of this book. Configuration of OpenBGPD on pfSense is straight forward for those with knowledge of BGP. During development of this package, we relied on [O'Reilly's BGP book](#) and recommend it for anyone looking to deploy BGP.

The general form of configuration for the OpenBGPD package is:

- Configure a group on the **Group** tab with the remote AS
- Configure one or more neighbors on the **Neighbors** tab as members of the defined **Group**
- Configure the **Settings** tab as desired for the local AS and networks to announce.

OSPF

An OSPF package using the [Quagga](#) routing daemon is also available. As with BGP, to install it:

- Navigate to **System > Package Manager**
- Click **Available Packages**
- Locate **Quagga_OSPF** in the list, or search for it
- Click the  **Install** to the right of the **Quagga_OSPF** package entry.
- Click  **Confirm**
- Wait for the installation to complete
- Navigate to **Services > Quagga OSPF**

OSPF is also a fairly complex routing protocol, though not as complex to setup as BGP can be. The details of configuring OSPFD are also outside the scope of this book, though for someone accustomed to OSPF the configuration options found in the GUI will be familiar.

The general form of configuration for the Quagga OSPF package is:

- Add interfaces as needed, with local interface subnets 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.

Route Troubleshooting

When diagnosing traffic flow issues, one of the first things to check is the routes known to pfSense.

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. 14.10: 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 14.3: 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.

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, 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.1.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  bbl-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.

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.

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.

BRIDGING

Creating a Bridge

In pfSense, bridges are added and removed at **Interfaces > (assign)** 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 > (assign)** 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.

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.

(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.

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).

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.

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.

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.

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.

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.

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.

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 in 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 > (assign)**
 - 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 > (assign)** 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 > (assign)**
- 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.

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.

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.

<code>net.link.bridge.pfil_member</code>	Packet filter on the member interface	1	
<code>net.link.bridge.pfil_bridge</code>	Packet filter on the bridge interface	0	

Fig. 15.1: 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.

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.

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.

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*.

Floating	WAN	LAN	WAN2	WAN3	WIFI	DMZ	IPsec				
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	 0/0 B	IPv4 UDP	0.0.0.0	68	255.255.255.255	67	*	none		Allow DHCP	  
☐	 0/626 KiB	IPv4 *	LAN net	*	*	*	*	none		Allow traffic on bridged interface	  

Fig. 15.2: 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.

Floating	WAN	LAN	WAN2	WAN3	WIFI	DMZ	IPsec				
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✔ 0/0 B	IPv4 UDP	0.0.0.0	68	255.255.255.255	67	*	none		Allow DHCP	  
☐	✔ 0/626 KiB	IPv4 *	LAN net	*	*	*	*	none		Allow traffic on bridged interface	  
☐	✔ 0/0 B	IPv6 UDP	fe80::/10	*	fe80::/10	546	*	none		Allow DHCPv6	  
☐	✔ 0/0 B	IPv6 UDP	fe80::/10	*	ff02::/16	546	*	none		Allow DHCPv6	  
☐	✔ 0/0 B	IPv6 UDP	fe80::/10	*	ff02::/16	547	*	none		Allow DHCPv6	  
☐	✔ 0/0 B	IPv6 UDP	ff02::/16	*	fe80::/10	547	*	none		Allow DHCPv6	  
☐	✔ 0/0 B	IPv6 UDP	fe80::/10	*	LAN address	546	*	none		Allow DHCPv6	  
☐	✔ 0/0 B	IPv6 *	LAN net	*	*	*	*	none		Allow traffic on bridged interface	  

Fig. 15.3: Firewall Rule to Allow both DHCP and DHCPv6

Bridging interoperability

Bridged interfaces are different from normal interfaces in some regards, thus there are a few features that are incompatible with bridging, and others where additional considerations must be made to accommodate bridging. This section covers features that work differently with bridging than with non-bridged interfaces.

Captive portal

Captive portal ([Captive Portal](#)) is not compatible with transparent bridging because it requires an IP on the interface being bridged, used to serve the portal contents, and that IP 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.

High Availability

High availability ([High Availability](#)) is not recommended with bridging at this time. Some 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.

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.

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.

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.

Normally each interface on pfSense 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](#).

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".

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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the May 2015 Hangout on Wireless Access Points which included practical examples of internal type bridges.

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.

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.

VIRTUAL LANS (VLANs)

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, pfSense, 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 [pfSense 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.

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.

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.

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.

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.

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.

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.

pfSense VLAN Configuration

This section covers how to configure VLANs on pfSense.

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 *igb0* 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

```

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>

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.

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 > (assign)** 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

- 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](#)

To assign the VLANs to interfaces:

Interfaces / VLANs / Edit

VLAN Configuration

Parent Interface

igb2 (00:08:a2:09:95:b1)

Only VLAN capable interfaces will be shown.

VLAN Tag

10

802.1Q VLAN tag (between 1 and 4094).

VLAN Priority

0

802.1Q VLAN Priority (between 0 and 7).

Description

DMZ

A group description may be entered here for administrative reference (not parsed).

Save

Fig. 16.1: Edit VLAN

Interface Assignments	Interface Groups	Wireless	VLANs	QinQs	PPPs	GREs	GIFs	Bridges	LAGGs
VLAN Interfaces									
Interface	VLAN tag	Priority	Description	Actions					
igb2	10		DMZ	 					
igb2	20		Phones	 					

Fig. 16.2: VLAN list

- Navigate to **Interfaces > (assign)**
- 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*

Interface Assignments		Interface Groups	Wireless	VLANs	QinQs	PPPs	GREs	GIFs	Bridges	LAGGs
Interface	Network port									
WAN	igb1 (00:08:a2:09:95:b6)									
LAN	igb0 (00:08:a2:09:95:b5)									
OPT1	VLAN 10 on igb2 (DMZ)									
OPT2	VLAN 20 on igb2 (Phones)									
Available network ports:	igb3 (00:08:a2:09:95:b2)									

Fig. 16.3: 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.

Switch VLAN Configuration

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 book.

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 pfSense 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.

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
```

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
```

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**.

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 16.1: 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. 16.4: VLAN Group Setting

- Select IEEE 802.1Q VLAN (Figure *Enable 802.1Q VLANs*).
- Click **OK** to confirm the switch to 802.1Q trunking, as shown in Figure *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*.

☒ IEEE 802.1Q VLAN
 ☐ Port-Based VLAN

ID	Description	Member
01	Default	01 02 03 04 05 06 07 08

Fig. 16.5: Enable 802.1Q VLANs

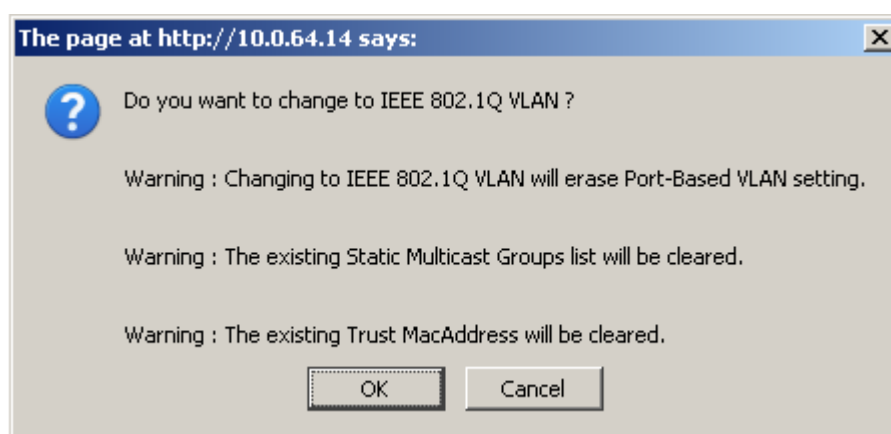


Fig. 16.6: Confirm change to 802.1Q VLAN

☒ IEEE 802.1Q VLAN
 ☐ Port-Based VLAN

VLAN Management :

Port	01	02	03	04	05	06	07	08
	U	U	U	U	U	U	U	U

☐ Not member
 ☒ T Tag egress packets
 ☒ U Untag egress packets

Fig. 16.7: 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](#).

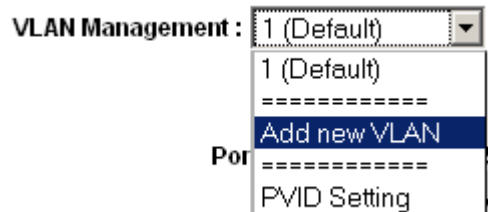


Fig. 16.8: Add New VLAN

- 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](#)).

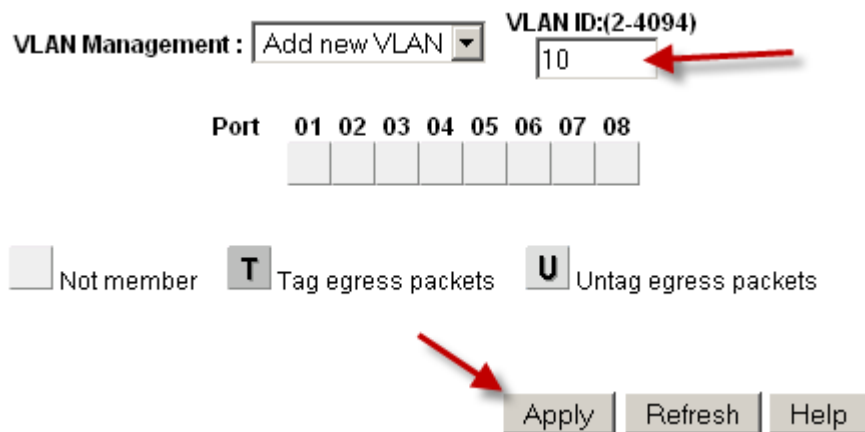


Fig. 16.9: Add VLAN 10

Add as many VLANs as needed, then continue to the next section.

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.

VLAN Management : Add new VLAN

VLAN ID:(2-4094) 20

Port 01 02 03 04 05 06 07 08

☐ Not member ☒ T Tag egress packets ☐ U Untag egress packets

Apply Refresh Help

Fig. 16.10: Add VLAN 20

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.

VLAN Management : 10 ☐ Remove VLAN

Port 01 02 03 04 05 06 07 08

☐ Not member ☒ T Tag egress packets ☐ U Untag egress packets

Fig. 16.11: 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.

Configure VLAN 20 membership

Select **20** from the VLAN Management drop down to configure the port memberships for VLAN **20**.

VLAN Management : 10 ☐ Remove VLAN

Port	01	02	03	04	05	06	07	08
	T	U	U	U				

☐ Not member ☒ Tag egress packets ☐ Untag egress packets

Fig. 16.12: Configure VLAN 10 Membership

VLAN Management : 20 ☐ Remove VLAN

Port	01	02	03	04	05	06	07	08
	T				U	U	U	U

☐ Not member ☒ Tag egress packets ☐ Untag egress packets

Fig. 16.13: 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*.

VLAN Management : 1 (Default)

Port

1 (Default)
10
20
=====

Add new VLAN
=====

PVID Setting

Fig. 16.14: 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.

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. 16.15: 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. 16.16: 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](#).

VLAN Management : 1 (Default)

☐ Remove VLAN

Port	01	02	03	04	05	06	07	08
	U							U

Fig. 16.17: 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.

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
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
```

pfSense 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.

QinQ is used to move groups of VLANs over a single link containing one outer tag, as can be found on some ISP, Metro Ethernet, or datacenter links between locations. It can be a quick/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.

Setting up QinQ interfaces on pfSense is fairly simple:

- Navigate to **Interfaces > (assign)**
- Click the **QinQ** tab
- Click  **Add** to add a new QinQ entry
- Configure the QinQ entry as follows:

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 by clicking



Add Tag, or in ranges such as 100-150

- Click **Save** to complete the interface

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.

QinQ Configuration

Parent interface igb3 (00:08:a2:09:95:b2)
Only QinQ capable interfaces will be shown.

First level tag 2000
This is the first level VLAN tag. On top of this are stacked the member VLANs defined below.

Option(s) ☒ Adds interface to QinQ interface groups
Allows rules to be written more easily.

Description To Site B
A description may be entered here for administrative reference (not parsed).

Member(s) Ranges can be specified in the inputs below. Enter a range (2-3) or individual numbers.
Click "Duplicate" as many times as needed to add new inputs.

Tag(s) 10 Delete

20 Delete

Save
+ Add Tag

Fig. 16.18: QinQ Basic Example

In Figure *QinQ List*, this entry is shown on the QinQ tab summary list.

Interface Assignments Interface Groups Wireless VLANs QinQs PPPs GREs GIFs Bridges LAGGs				
QinQ Interfaces				
Interface	Tag	QinQ members	Description	Actions
igb3	2000	10 20	To Site B	
				+ Add

Fig. 16.19: QinQ 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.

Interface Assignments	Interface Groups	Wireless	VLANs	QinQs	PPPs	GREs	GIFs	Bridges	LAGGs

Interface Groups			
Name	Members	Description	Actions
QinQ	igb3_2000_10, igb3_2000_20, igb3_2000	QinQ VLANs group	 
 Add			

Fig. 16.20: QinQ Interface Group

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_vlan10` 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.

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.

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).

MULTIPLE WAN CONNECTIONS

Multi-WAN Terminology and Concepts

This section covers the terminology and concepts necessary to understand to deploy multi-WAN with pfSense.

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. Do not select a gateway on the **Interfaces** menu entry for local interfaces. 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. For example, interfaces with a gateway set have `reply-to` on their firewall rules, they are used as exit interfaces for automatic and hybrid outbound NAT, and they are treated as WANs by the traffic shaper wizard.

Note: 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**.

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.

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.

Failover

Failover refers to the ability to use only one WAN connection, but switch to another WAN if the preferred connection fails. This is useful for situations where certain traffic, all 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](#).

Load Balancing

The **Load Balancing** functionality in pfSense allows traffic to be distributed over multiple WAN connections in a round-robin fashion. This is done 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.

Monitor IP Addresses

When configuring failover or load balancing, each gateway is associated with a monitor IP address (*Monitor IP*). In a typical configuration, pfSense will ping this IP address and if it stops responding, the interface 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 fail, the interface 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](#). The thresholds for packet loss, latency, down time, and even the probing interval of the gateway are all individually configurable.

State Killing/Forced Switch

When a gateway has failed, pfSense 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, enabled by default. For information on changing this setting, see [Gateway Monitoring](#).

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. When **Default Gateway Switching** (*Default Gateway Switching*) is enabled, the default gateway for the firewall will be switched to the next available gateway if the normal default gateway fails, and then switched back when that WAN recovers.

Policy Routing, Load Balancing and Failover Strategies

This section provides guidance on common Multi-WAN goals and how they are achieved with pfSense.

Bandwidth Aggregation

One of the primary desires with multi-WAN is bandwidth aggregation. With load balancing, pfSense can help accomplish this goal. There is, however, one caveat: If the firewall has two 5 Mbps WAN circuits, it cannot get 10 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, 10 Mbps of throughput can only be achieved using two 5 Mbps circuits when multiple connections are involved. Applications that utilize multiple connections, such as many 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.

Segregation of Priority Services

In some situations, a site may have 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.

Failover Only

There are scenarios where only using failover is the best practice. Some pfSense users have a secondary backup Internet connection with a low bandwidth limit such as a 3G 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.

Unequal Cost Load Balancing

pfSense can achieve unequal cost load balancing by setting appropriate weights on the gateways as discussed in [Weight](#). 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 17.1: 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%
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 necessarily 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.

Multi-WAN Caveats and Considerations

This section contains the caveats and considerations specific to multi-WAN in pfSense.

Multiple WANs sharing a single gateway IP

Because of 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 pfSense sees each WAN gateway as a unique IP address.

When using the NAT device, it can be configured to forward all traffic back to pfSense 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 [Monitor IP](#)).

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 pfSense. Configuration of MLPPP is covered in [Multi-Link PPPoE \(MLPPP\)](#).

Local Services and Multi-WAN

There are some 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 system's 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.

In the case of traffic initiated on the Internet destined for any WAN interface, pfSense automatically uses pf's *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 **Default Gateway Switching** to work properly with Multi-WAN. See [Default Gateway Switching](#) 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 OPT WAN interface, as described later in this chapter. There are no other caveats to DNS forwarder in multi-WAN environments.

DynDNS

DynDNS entries can be set using a gateway group for their interface. This will move a DynDNS 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 [Multi-WAN with HA](#).

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 [Multi-WAN for IPv6](#).

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
- Configure the DNS Resolver or Forwarder for Multi-WAN, starting by setting at least one unique DNS server for each WAN gateway under **System > General Setup**
- Use the gateway group on LAN firewall rules

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 very 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

Configuring a Gateway Group for Load Balancing or Failover

To create a gateway group for Load Balancing or Failover:

- Navigate to **System > Routing, Groups** tab
- Click  **Add** to create a new gateway group
- Fill in the options on the page as needed:

Group Name A name for the 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.

Tier Choose the priority for gateways within the group. Inside gateway groups, gateways are arranged in Tiers. Tiers are numbered 1 through 5, and **lower** numbers are used **first**. For example, gateways on *Tier 1* are used before gateways on *Tier 2*, and so on. See the next sections for more detail on how to use Tiers.

Virtual IP Optionally specifies a virtual IP address to use for an interface, if one exists. This option is used for features such as OpenVPN, allowing a specific virtual address to be chosen, rather than using only the Interface address directly when a specific gateway is active in the group. In most cases, this is left at the default value *Interface Address*.

Trigger Level Decides when to mark a gateway as down.

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 *Packet Loss Thresholds*).

High Latency Marks the gateway as down when latency crosses the lower alert threshold (See *Latency Thresholds*).

Packet Loss or High Latency Marks the gateway as down for either type of alert.

Description Text describing the purpose of this gateway group.

- 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 *Weight*.

Failover

Gateways on a **lower** number tier are preferred, 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, many 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.

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. The suggested means of working around this 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.

Interface and DNS Configuration

The first two items to configure for Multi-WAN are Interfaces and DNS.

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.

DNS Server Configuration

If the DNS Forwarder is in use, or if the DNS Resolver will be used in **forwarding** mode, pfSense 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 DNS servers from only a single WAN are used, an outage of that WAN connection will result in a complete Internet outage regardless of policy routing configuration since DNS will no longer function.

DNS Resolver Configuration

The DNS Resolver can work with Multi-WAN but the exact configuration depends on the desired behavior and current settings.

If DNSSEC must be used and the configured DNS servers do not support DNSSEC, then forwarding mode cannot be enabled. This can still function with Multi-WAN but requires **Default Gateway Switching**. See [Default Gateway Switching](#).

If DNSSEC is not a requirement for this firewall, or the configured DNS servers support DNSSEC, 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** if the configured upstream DNS servers do not support DNSSEC

DNS Servers and Static Routes

When using the DNS Forwarder or the DNS Resolver in forwarding mode, pfSense 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 so pfSense will 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 (up to four).
- 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 pfSense 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, **Default Gateway Switching** is required so that the `unbound` daemon can maintain outbound connectivity.

Scaling to Large Numbers of WAN Interfaces

There are numerous pfSense users deploying 6-12 Internet connections on a single installation. One pfSense 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 uses pfSense to load balance 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.

Multi-WAN and NAT

The default NAT rules generated by pfSense will translate any traffic leaving a WAN-type interface to that interface IP address. 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 an WAN-type interface to that interface IP address. This is all handled automatically unless Manual Outbound NAT is enabled.

Policy routing firewall rules direct the traffic to the WAN interface used, and the Outbound and 1:1 NAT rules specify how the traffic will be translated as it leaves that WAN.

Multi-WAN and Manual Outbound NAT

If **Manual Outbound NAT** must be used with multi-WAN, ensure NAT rules are configured for all WAN-type interfaces.

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](#).

Multi-WAN and 1:1 NAT

1:1 NAT entries are specific to a single WAN interface and, like outbound NAT, only controls what happens to traffic as it leaves 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.

Policy Routing Configuration

At this point, the firewall is prepared for Multi-WAN but it will not yet be used. Traffic will not fail over or be load balanced without policy routing firewall rules in place.

Note: One possible exception is if **Default Gateway Switching** is enabled ([Default Gateway Switching](#)), in which case failover could still function without policy routing.

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.

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. 17.1: Bypass Policy Routing Example Rules

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.

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.

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.

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 *Monitor IP*.

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, 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 pfSense 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 *Monitor IP*.

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 be marked as down.

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 pfSense 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 spammy ads, so we provide a couple sites that simply report the client IP address:

- <http://www.pfsense.org/ip.php>
- <https://www.pfsense.org/ip.php>
- <https://portal.pfsense.org/ip.php>

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.

If the IP address never changes, try several different sites, and make sure the browser really is requesting the page again, and not returning something from its cache or using a persistent connection to the server. Manually deleting the cache, closing and reopening the browser, and trying multiple web browsers are good things to attempt before troubleshooting the load balancer configuration further.

Using curl as described in *Verifying load balancing* is a better test as it ensures cache and persistent connections will have no impact on the results.

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 simply directed on a round robin basis without regard for bandwidth usage.

Troubleshooting

This section describes some of the most common problems with multi-WAN and how to troubleshoot them.

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.

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.

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.

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 as described in [Verifying load balancing](#).
- Check that the traffic is not using a proxy or otherwise being initiated from a daemon on the firewall itself.

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 [Data Payload](#) 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 `tracert` to make sure the traffic is leaving using the expected path.

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.

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 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.

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 [Connecting with 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.

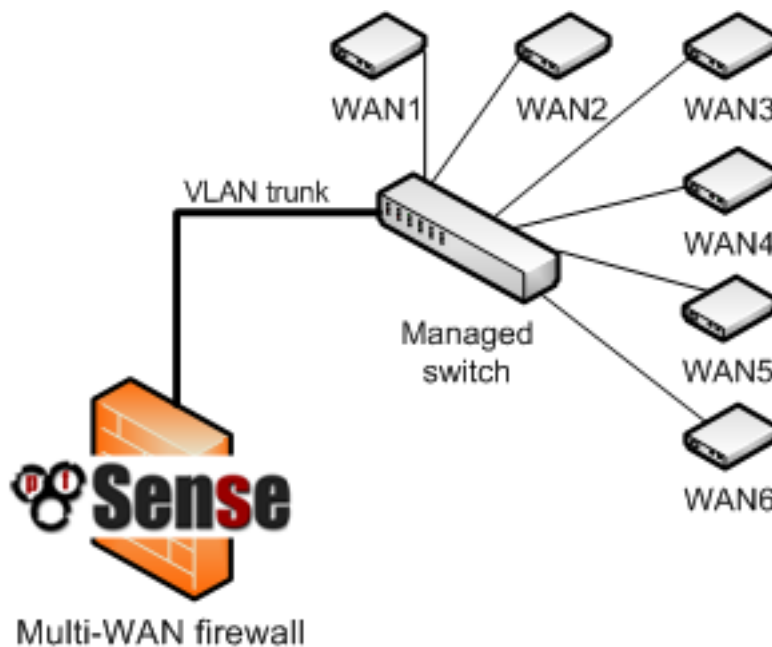


Fig. 17.2: Multi-WAN on a stick

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.

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

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.

Alternate Tactics

Some users prefer to configure LAN with a “private” IPv6 subnet from the `fc00::/7` space and setup NPt for both WANs.

Multi-Link PPPoE (MLPPP)

Multi-Link PPPoE (MLPPP) is a unique WAN option that can bond 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 5 Mbit/s DSL lines in a bundle, it could potentially get 15Mbit/s from a single connection.

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 pfSense.

Setup

Setup for MLPPP is very 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**

pfSense will then attempt to bond the lines using MLPPP.

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.

The multiple WAN (multi-WAN) capabilities of pfSense allow a firewall to utilize multiple Internet connections to achieve more reliable connectivity and greater throughput capacity. Before proceeding with a multi-WAN configuration, the firewall must have a functional two interface (LAN and WAN) configuration. pfSense is capable of handling many WAN interfaces, with multiple deployments using 10-12 WANs in production. It will scale even higher than that, though we aren't aware of any installations using more than 12 WANs.

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 chapter starts by covering items to consider when implementing *any* multi-WAN solution, then covers multi-WAN configuration with pfSense.

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.

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 very important to make sure connectivity choices for a multi-WAN deployment 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 pfSense 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.

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.

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 stick with DS1 or DS3 connectivity. In areas where multiple lower cost broadband options are available, such as DSL and cable, the combination of pfSense 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.

VIRTUAL PRIVATE NETWORKS

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.

Interoperability

To interoperate with a firewall or router product from another vendor, 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 is interoperable with a few other packaged firewall/VPN solutions, but not many. Interoperability in this sense isn't applicable with other VPN types since they are not intended for site-to-site applications.

Authentication considerations

In current versions of pfSense, all VPN types support user authentication. 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.

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 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. pfSense includes 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.

IPsec and OpenVPN are preferable options in many scenarios for other reasons discussed throughout this chapter.

Multi-WAN capable

If users require the ability to connect to multiple WANs, both IPsec and OpenVPN are capable of handling such configurations.

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, 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 pfSense 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. A good free option for Windows is the [Shrew Soft client](#). 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 by pfSense and are covered in [IPsec](#).

The Cisco-style IPsec client included with OS X and iOS devices is fully compatible with pfSense IPsec using xauth. Configuration for the iOS client is covered in [iOS 9 IKEv2 Client Configuration](#).

Many Android phones also include a compatible IPsec client, which is discussed in [Android strongSwan IKEv2 Client Configuration](#).

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 4.x and later devices can use a freely available OpenVPN client that works well and doesn't require rooting the device. That client is covered in [Android 4.x and later](#). Older versions of Android may also be able to use OpenVPN via an alternate client. There are other options available if the device is rooted, but that is beyond the scope of this book.

iOS also has a native OpenVPN client. For more information, see [iOS](#).

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 the most firewall-friendly of the VPN options. Since it uses TCP or UDP and is not affected by any 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.

Cryptographically secure

One of the critical functions of a VPN is to ensure the confidentiality of the data transmitted.

IPsec 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.

Recap

Table *Features and Characteristics by VPN Type* shows an overview of the considerations provided in this section.

Table 18.1: Features and Characteristics by VPN Type

VPN Type	Client included in most OSes	Widely interoperable	Multi-WAN	Cryptographically secure	Firewall friendly
IPsec	Varies by mode	Yes	Yes	Yes	No (without NAT-T)
Open-VPN	No	No	Yes	Yes	Yes

VPNs and Firewall Rules

VPNs and firewall rules are handled somewhat inconsistently in pfSense. 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.

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**.

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 on pfSense. 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.

VPNs and IPv6

There are some special considerations for VPNs when using them in combination with IPv6. The two main items of concern are:

- Whether or not a certain VPN type supports IPv6
- Making sure the firewall rules don't allow unencrypted traffic in that should be coming over a VPN.

IPv6 VPN Support

Support for IPv6 varies from type to type and in client support. Be sure to check with the vendor of the other device in order to make sure a non-pfSense firewall or client supports IPv6 VPNs.

IPsec

pfSense supports IPsec using IKEv1 over IPv6 with one quirk: If an IPv6 peer address is used, the tunnel can only carry IPv6 phase 2 networks, and the same for IPv4. Traffic cannot be mixed between address families. See [IPsec and IPv6](#).

When an IPsec tunnel is set for IKEv2, it can include both IPv4 and IPv6 Phase 2 definitions concurrently.

OpenVPN

OpenVPN fully supports IPv6 for site-to-site and mobile clients, and tunnels can carry both IPv4 and IPv6 traffic concurrently. See [OpenVPN and IPv6](#).

IPv6 VPN and Firewall Rules

As mentioned briefly in [Firewall and VPN Concerns](#), some special care must be taken when routing IPv6 traffic across a VPN and using publicly routable subnets. The same advice would also apply 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](#).

VPNs provide a means of tunneling traffic through an encrypted connection, preventing it from being seen or modified in transit. pfSense offers three VPN options: IPsec, OpenVPN, and L2TP. This chapter provides an overview of VPN usage, the pros and cons of each type of VPN in pfSense, and how to decide which is the best fit for a particular environment. Subsequent chapters 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 some other 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.

PPTP Warning

PPTP server support has been removed from pfSense. 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; Any system that handles PPTP is no longer secure. The reason for the insecurity is that 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 such as OpenVPN or IPsec 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/>.

Common deployments

There are four common uses of the VPN capabilities of pfSense, each covered in this section.

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.

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.

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](#).

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.

IPSEC

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.

Choosing configuration options

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.

Phase 1 Settings

The settings here control the phase 1 negotiation portion of the tunnel, as described previously.

Enable/Disable Tunnel

The **Disabled** checkbox controls whether or not this tunnel (and its associated phase 2 entries) are active and used.

Key Exchange Version

The **Key Exchange Version** selector controls whether the tunnel will use IKE version 1 (V1) or IKE version 2 (V2). IKEv2 is a newer version of IKE that is desirable in many ways. The differences are discussed in [IKE](#). In most cases, IKEv1 will be used unless both sides properly support IKEv2.

Internet Protocol

The **Internet Protocol** selector sets 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 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 pfSense firewall, 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.

Remote Gateway

The **Remote Gateway** is the IPsec peer for this phase 1. This is the endpoint on the other side of the tunnel to which IPsec will negotiate this phase 1. 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.

Description

The **Description** for the phase 1 is some text to use for identifying this phase 1. It's not used in the IPsec settings, it's only for reference.

Authentication Method

An IPsec phase 1 can be authenticated using a pre-shared key (PSK) or RSA 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

When using *Mutual PSK*, the peer is validated using a defined string. The longer the better, but 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 RSA

In *Mutual RSA* mode, 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 their shared CA. The CA and certificate must be created for the tunnel before attempting to setup the phase 1.

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 RSA+Xauth

Used with mobile IPsec and IKEv1, this selection enables xauth username and password verification along with RSA certificate authentication using certificates on both the client and server.

Hybrid RSA+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 RSA+Xauth*, but it is easier on the clients.

EAP-TLS

Used with mobile IPsec and IKEv2, RSA EAP-TLS verifies that certificates on the client and server are from the same shared CA, similar to Mutual RSA. The client and server certificates require special handling:

- The server certificate must have the firewall's name as it exists in DNS listed in its Common Name, and again as a Subject Alternative Name (SAN). The firewall's IP address must also be listed in a SAN.
- The identifier in Phase 1 must also be set to match the firewall's hostname as listed in the Common Name of the certificate.
- The client certificate must have the user's name 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

For IKEv1, two **Negotiation Mode** choices are supported: *main*, *aggressive*. This selection is not present when using IKEv2.

Main Mode

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.

Aggressive Mode

Aggressive is generally the most compatible and is the fastest mode. It is a bit more forgiving with identifier types, and tends to be more successful when negotiating with third-party devices in some cases. 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.

My identifier / Peer Identifier

Here, choose the identifier used to send to the remote peer, and also for verifying the identity of the remote peer. The following identifier types can be chosen for the **My Identifier** and **Peer Identifier** selectors. If needed, a text box will appear to enter a value to be used for the identifier.

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 the packets were received, which should be the **Remote Gateway**.

IP Address

The *IP Address* option allows a different 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, 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`, rather than an FQDN.

ASN.1 Distinguished Name

If using *Mutual RSA* 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 not relevant or available for a **Peer Identifier** as the hostname may be used directly in the **Remote Gateway** field and use *Peer IP Address* for the 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 (If using Mutual PSK)

This field is used to enter the PSK for phase 1 authentication. As mentioned previously, make this a long/complex key. If this PSK has been provided by the peer, enter it here. If a new PSK must be generated, we recommend using a password generation tool set to a length of at least 15, but it can be much longer.

Phase 1 Encryption algorithms

There are many options for encryption algorithms on both phase 1 and phase 2.

The current options are all considered cryptographically secure. Which to choose depends on the device to which the tunnel will connect, and the hardware available in this firewall. Generally speaking, AES is the most desirable cipher and the longest key length (256 bits) is best. When connecting to third party devices, 3DES (also called “Triple DES”) is a common choice as it may be the only option the other end supports.

More information about ciphers and acceleration is available in *Phase 2 Encryption algorithms*.

Phase 1 Hash algorithms

Hash algorithms are used with IPsec to verify the authenticity of packet data. MD5, SHA1, SHA256, SHA384, SHA512, and AES-XCBC are the available hash algorithms on phase 1 and phase 2. All are considered cryptographically secure, though SHA1 (Secure Hash Algorithm, Revision 1) and its variants are considered the stronger than MD5. SHA1 does require more CPU cycles than MD5, and the larger values of SHA in turn require even greater CPU power. 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.

Note: The implementation of SHA256-512 is [RFC 4868](#) compliant on the FreeBSD version used by pfSense. RFC 4868 compliance breaks compatibility with stacks that implemented [draft-ietf-ipsec-ciph-sha-256-00](#), including FreeBSD 8.1 and earlier. Before using SHA256, 384, or 512, check with the other side to ensure they are also RFC 4868 compliant implementations or they will not work. The relevant [FreeBSD commit message](#) when this happened explains in a little more detail.

DH key group

All of the DH (Diffie-Hellman, named after its authors) key group options are considered cryptographically secure, though the higher numbers are slightly more secure at the cost of increased CPU usage.

Lifetimes

The lifetime specifies how often the connection must be rekeyed, specified in seconds. 28800 seconds on phase 1 is a common configuration and is appropriate for most scenarios.

My Certificate (If using Mutual RSA)

This option only appears if using an RSA-based **Authentication Mode**. The list is populated using the certificates present in the firewall’s configuration. Certificates can be imported and managed under **System > Cert Manager** on the **Certificates** tab. Choose the certificate to use for this IPsec phase 1 from the list. The CA for this certificate must match the one chosen in the **My Certificate Authority** selector.

My Certificate Authority (If using Mutual RSA)

This option only appears if using an RSA-based **Authentication Mode**. The list is populated using the CAs present in the firewall's configuration. A CA can be imported and managed under **System > Cert Manager**. Choose the CA to use for this IPsec phase 1 from the list.

Disable Rekey

Selecting this option will instruct pfSense to not initiate a rekey event on the tunnel. Some clients (Especially Windows clients behind NAT) will misbehave when they receive a rekey request, so it is safer in these cases to allow the client to initiate the rekey by disabling the option on the server. Normally both sides would rekey as needed, but if the tunnel often fails when a rekey event occurs, try selecting this option on only one side.

Disable Reauth

This option only appears for IKEv2 tunnels, IKEv1 will always re-authenticate. If this option is checked, then when a tunnel rekeys it does not re-authenticate the peer. When unchecked, the SA is removed and negotiated in full rather than only rekeying.

Responder Only

If **Responder Only** is selected, then pfSense will not attempt to initiate the tunnel when traffic attempts to cross. The tunnel will only be established when the far side initiates the connection. 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.

NAT Traversal

The NAT Traversal option, also known as NAT-T, is only available for IKEv1. IKEv2 has NAT Traversal integrated in such a way that the option is unnecessary. NAT Traversal can encapsulate the 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 will be behind a NAT device, then NAT Traversal will likely be necessary. The default setting of *Auto* will use NAT Traversal in cases where its need is detected. The option may also be set to *Force* to ensure NAT Traversal will always be used for the tunnel. This can help if there is a known issue carrying ESP traffic between the two endpoints.

MOBIKE

MOBIKE is an extension to IKEv2 that handles multi-homed clients and clients that roam between different IP addresses. This is primarily used with mobile clients to allow them to switch remote addresses while keeping the connection active.

Split Connections

This option is specific to IKEv2 and configures the Phase 2 entries such a way they use separate connection entries, rather than one single traffic selector per child Security Association. Specifically, this is known to be an issue with Cisco products such as ASA.

If an IKEv2 tunnel is in use with multiple Phase 2 entries, and only one Phase 2 network pair will establish a connection, activate this option.

Dead Peer Detection (DPD)

Dead Peer Detection (DPD) is a periodic check that the host on the other end of the IPsec tunnel is still alive. If a DPD check fails, the tunnel is torn down by removing its associated SAD entries and renegotiation is attempted.

The **Delay** field controls how often a DPD check is attempted, and the **Max Failures** field controls how many of these checks must fail before a tunnel is considered to be a down state. The default values of 10 seconds and 5 failures will result in the tunnel being considered down after approximately one minute. These values may be increased for bad quality links to avoid tearing down a usable, but lossy, tunnel.

Phase 2 Settings

The phase 2 settings for an IPsec tunnel govern what traffic will enter the tunnel as well as how that traffic is encrypted. For normal tunnels, this controls the subnets that will enter the firewall. For mobile clients this primarily controls the encryption for phase 2, but can also optionally supply a list of networks to the clients for use in split tunneling. Multiple phase 2 definitions can be added for each phase 1 to allow using multiple subnets inside of a single tunnel.

Enable/Disable

This setting controls whether or not this phase 2 entry is active.

Mode

This option allows traditional tunneling mode of IPsec, or transport mode. Tunneling mode can also specify either IPv4 or IPv6.

Tunnel IPv4/IPv6 Mode

When using either *Tunnel IPv4* or *Tunnel IPv6* for this phase 2 entry, the firewall will tunnel IPv4 or IPv6 traffic matching the specified **Local Network** and **Remote Network**. A phase 2 can be for either IPv4 or IPv6, and the network values are validated based on that choice. Traffic matching both the Local Network and Remote Network will enter the tunnel and be delivered to the other side.

Note: With IKEv1, only one of IPv4 or IPv6 may be used, and it must match the same address family used to establish the tunnel's Phase 1. With IKEv2, both types may be used in the same tunnel.

Transport Mode

Transport mode will encrypt traffic between the IP addresses used as the phase 1 endpoints. This mode allows encrypting traffic from the firewall's external IP address to the external IP address on the far side. Any traffic sent between the two nodes will be encrypted, so using other tunneling methods that do not employ encryption, such as a GIF or GRE tunnel, can be safely used. The **Local Network** and **Remote Network** are not set for transport mode, it assumes the addresses based on the phase 1 settings.

Local Network (If using a Tunnel mode)

As the name implies, this option sets the **Local Network** which will be associated with this phase 2. 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 subnet choices for each interface (e.g. *LAN subnet*), as well as *Address* and *Network* choices that allow entering an IP address or subnet manually.

NAT/BINAT Translation

To perform NAT on local network addresses to make them appear as a different subnet or as a public IP address, use the **NAT/BINAT Translation** fields. If a single IP address is specified in **Local Network** and a single IP address in the **NAT/BINAT Translation Type** field, then a 1:1 NAT translation will be set up between the two. 1:1 NAT is also setup if a subnet of the same size is used in both fields. If the Local Network is a subnet, but the **NAT/BINAT Translation** 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 in the NAT field. If NAT is not needed on the IPsec traffic, leave it set to *None*.

Remote Network (If using a Tunnel mode)

This option (only present for non-mobile tunnels) specifies the IP *Address* or *Network* that exists on the other (remote) side of the VPN. It operates similarly to the **Local Network** option mentioned previously.

Protocol

IPsec has the option of choosing AH (Authenticated Header) or ESP (Encapsulating Security Payload). In nearly all circumstances, ESP is used as it is the only option that encrypts traffic. AH only provides assurance the traffic came from the trusted source and is rarely used.

Phase 2 Encryption algorithms

In systems with AES-NI, the fastest and most secure choice is AES-GCM, provided the remote device supports it as well. When using AES-GCM in Phase 2, use AES in Phase 1 with an equivalent key length. Also if AES-CGM is used, do not select any options for **Hash Algorithms** in Phase 2.

When using systems with *glxb* accelerators, such as ALIX, choose AES 128 for best performance. For systems with *hifn* accelerators, chose 3DES or AES for best performance. Both AES and Blowfish allow selecting the key length of the cipher in varying steps between 128-bit and 256-bit. Lower values will be faster, larger values are more cryptographically secure. For systems without a hardware cryptography accelerator, Blowfish and CAST are the fastest options.

The phase 2 encryption choices allow for multiple selections so that either multiple choices will be accepted when acting as a responder, or multiple combinations will be tried when working as an initiator. It's best to only select the single desired cipher, but in some cases selecting multiple will allow a tunnel to work better in both a responder and initiator role.

Phase 2 Hash algorithms

As with the Encryption Algorithms, multiple hashes may be selected. It is still best to only select the single desired choice if possible. For more discussions on the quality of the various hash types, see *Phase 1 Hash algorithms*.

Note: When using AES-GCM for the **Phase 2 Encryption Algorithm**, do not select **any** options for the **Hash algorithm**!

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 when rekeying occurs. 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.

Lifetime

The **Lifetime** option specifies how often the connection must be rekeyed, in seconds. 3600 seconds on phase 2 is a common configuration and is appropriate for most scenarios.

Automatically Ping Host (Keep Alive)

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**.

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).

Site-to-Site

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.

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. An IP address from the remote subnet to ping is optional, but recommended to keep the tunnel alive. The firewall doesn't check for replies, as any traffic initiated to an IP address on the remote network will trigger IPsec negotiation, so it doesn't matter if the host actually responds or not as long as it is an IP address on the other side of the connection. Aside from the cosmetic tunnel **Description** and these pieces of information, the other connection settings will be identical.

In this example and some of the subsequent examples in this chapter, the following settings will be assumed:

Table 19.1: 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

Start with Site A. Create the tunnel by clicking  **Add P1**. The phase 1 configuration page for the tunnel is shown. Many of these settings may be left at their default values.

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 IKEv1 or IKEv2. For this example, IKEv2 is used, but if one side does not support IKEv2, use IKEv1 instead.

Internet Protocol Will be *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 Some text to state the purpose or identity of the tunnel. It is a good idea to put the name of Site B in this box, and some detail about the tunnel's purpose to help with future administration. For this example *ExampleCo London Office* is used for the **Description** to identify where the tunnel terminates.

Tunnels
Mobile Clients
Pre-Shared Keys
Advanced Settings

General Information

Disabled
☐ Set this option to disable this phase1 without removing it from the list.

Key Exchange version
V2
Select the Internet Key Exchange protocol version to be used. Auto uses IKEv2 when initiator, and accepts either IKEv1 or IKEv2 as responder.

Internet Protocol
IPv4
Select the Internet Protocol family.

Interface
WAN
Select the interface for the local endpoint of this phase1 entry.

Remote Gateway
203.0.113.5
Enter the public IP address or host name of the remote gateway

Description
ExampleCo London Office
You may enter a description here for your reference (not parsed).

Fig. 19.1: Site A VPN Tunnel Settings

The next section controls IPsec phase 1, or Authentication. It is shown in Figure [Site A Phase 1 Settings](#). 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.

Peer Identifier The default, *Peer IP Address*, is kept.

Pre-Shared Key This is the most important setting to get correct. As mentioned in the VPN overview, IPsec 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. The same exact key will need to 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 like `aBc123%XyZ9$7qwErtY99`.

Encryption Algorithm Use *AES* with a key length of *256 bits*.

Hash Algorithm Use *SHA256* if both sides support it, otherwise use the default *SHA1*.

DH Group The default of *2 (1024 bit)* is OK, higher values are more secure but use more CPU.

Lifetime May also be specified, otherwise the default value of *28800* will be used.

Disable rekey Leave unchecked

Responder only Leave unchecked

NAT Traversal Leave on *Auto*, since in this example neither firewall is behind NAT.

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** to complete the phase 1 setup.

After the phase 1 has been added, add 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](#).

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.

Mode Since tunneling traffic is desired, select *Tunnel IPv4*

Local Subnet Best to leave this as *LAN Subnet*, but it could also be changed to *Network* with the proper subnet value filled in, in this case *10.3.0.0/24*. Leaving it as *LAN Subnet* will ensure that if the network is renumbered, this end of the tunnel will follow. If that does happen, the other end must be changed manually.

NAT/BINAT Set to *None*.

Remote Subnet Set to the network at Site B, in this case *10.5.0.0/24*.

The remainder of the phase 2 settings, seen in Figure [Site A 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.

Protocol Set to *ESP* for encryption.

Phase 1 Proposal (Authentication)

Authentication Method

Mutual PSK

Must match the setting chosen on the remote side.

My identifier

My IP address

Peer identifier

Peer IP address

Pre-Shared Key

aBc123%XyZ9\$7qwErty99

Enter your Pre-Shared Key string.

Phase 1 Proposal (Algorithms)

Encryption Algorithm

AES

256 bits

Hash Algorithm

SHA256

Must match the setting chosen on the remote side.

DH Group

2 (1024 bit)

Must match the setting chosen on the remote side.

Lifetime (Seconds)

28800

Advanced Options

Disable rekey

☐

Disables renegotiation when a connection is about to expire.

Disable Reauth

☐

Whether rekeying of an IKE_SA should also reauthenticate the peer. In IKEv1, reauthentication is always done.

Responder Only

☐

Enable this option to never initiate this connection from this side, only respond to incoming requests.

MOBIKE

Disable

Set this option to control the use of MOBIKE

Split connections

☐

Enable this to split connection entries with multiple phase 2 configurations. Required for remote endpoints that support only a single traffic selector per child SA.

Dead Peer Detection

☒

Enable DPD

Delay

10

Delay between requesting peer acknowledgement.

Max failures

5

Number of consecutive failures allowed before disconnect.

Fig. 19.2: Site A Phase 1 Settings

IPsec Tunnels							
	IKE	Remote Gateway	Mode	P1 Protocol	P1 Transforms	P1 Description	Actions
☐	Disable	V2	WAN 203.0.113.5	AES (256 bits)	SHA256	ExampleCo London Office	
Show Phase 2 Entries (0)							

Fig. 19.3: Site A Phase 2 List (Empty)

IPsec Tunnels								
	IKE	Remote Gateway	Mode	P1 Protocol	P1 Transforms	P1 Description	Actions	
☐	Disable	V2	WAN 203.0.113.5	AES (256 bits)	SHA256	ExampleCo London Office		
Mode Local Subnet Remote Subnet P2 Protocol P2 Transforms P2 Auth Methods P2 actions Add P2								

Fig. 19.4: Adding a Phase 2 entry to Site A

General Information	
Disabled	☐ Disable this phase 2 entry without removing it from the list.
Mode	Tunnel IPv4
Local Network	LAN subnet Type Address / 0
NAT/BINAT translation	None Type Address / 0 If NAT/BINAT is required on this network specify the address to be translated
Remote Network	Network Type 10.5.0.0 / 24 Address
Description	ExampleCo London LAN You may enter a description here for your reference (not parsed).

Fig. 19.5: Site A Phase 2 General Settings

Encryption algorithm Ideally, if both sides support it, select *AES256-GCM* with a *128 bit* key length. Otherwise, use *AES 256*, or whichever cipher both ends will support.

Hash algorithm With AES-GCM in use, no hash is required. Otherwise, use *SHA 256* or *SHA 1*. Avoid *MD5* when possible.

PFS Perfect Forward Secrecy can help protect against certain key attacks, but is optional. In this example, it is disabled.

Lifetime Use 3600 for this example.

Phase 2 Proposal (SA/Key Exchange)	
Protocol	ESP ESP is encryption, AH is authentication only.
Encryption Algorithms	☐ AES Auto ☐ AES128-GCM Auto ☐ AES192-GCM Auto ☒ AES256-GCM 128 bits ☐ Blowfish Auto ☐ 3DES ☐ CAST128 Use 3DES for best compatibility or if you have a hardware crypto accelerator card. Blowfish is usually the fastest in software encryption.
Hash Algorithms	☐ MD5 ☐ SHA1 ☐ SHA256 ☐ SHA384 ☐ SHA512 ☐ AES-XCBC
PFS key group	off
Lifetime	3600 Seconds

Fig. 19.6: Site A Phase 2 Settings

Lastly, an IP address can be entered for a system on the remote LAN that will be periodically sent an ICMP ping, as in Figure *Site A Keep Alive*. The return value of the ping is not checked, this will only send traffic the tunnel so that it will stay established. In this setup, the LAN IP address of the pfSense firewall at Site B, *10.5.0.1*, is used.

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*.

Advanced Configuration	
Automatically ping host	10.5.0.1
	IP Address

Fig. 19.7: Site A Keep Alive

The IPsec tunnel configuration has been changed. You must apply the changes in order for them to take effect.	✓ Apply Changes

Fig. 19.8: Apply IPsec Settings

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. These rules must be added to the **IPsec** tab under **Firewall Rules**. 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). In each case, make sure the Source address(es) are Site B addresses, such as 10.5.0.0/24. The destination addresses will be the Site A network, 10.3.0.0/24.

Now that Site A is configured, it is time to tackle Site B. Repeat the process on Site B's 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 Settings* and Figure *Site B Phase 2 Settings*:

- The phase 1 settings for WAN address and Description
- The phase 2 tunnel networks
- The keep alive setting

Add a Phase 1 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.

- 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.

Automatically ping host (Figure *Site B Keep Alive*). The LAN IP address of the pfSense firewall at Site A, 10.3.0.1.

- 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.

Both tunnels are now configured and are now 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. Since the local network includes an address that the firewall

has, a connect button is offered on this screen that will initiate a ping to the remote phase 2. 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

General Information	
Disabled	☐ Set this option to disable this phase1 without removing it from the list.
Key Exchange version	V2 Select the Internet Key Exchange protocol version to be used. Auto uses IKEv2 when initiator, and accepts either IKEv1 or IKEv2 as responder.
Internet Protocol	IPv4 Select the Internet Protocol family.
Interface	WAN Select the interface for the local endpoint of this phase1 entry.
Remote Gateway	198.51.100.3 Enter the public IP address or host name of the remote gateway
Description	ExampleCo Austin Office You may enter a description here for your reference (not parsed).

Fig. 19.9: Site B Phase 1 Settings

General Information	
Disabled	☐ Disable this phase 2 entry without removing it from the list.
Mode	Tunnel IPv4
Local Network	LAN subnet / 0 Type Address
NAT/BINAT translation	None / 0 Type Address If NAT/BINAT is required on this network specify the address to be translated
Remote Network	Network 10.3.0.0 / 24 Type Address
Description	ExampleCo Austin LAN You may enter a description here for your reference (not parsed).

Fig. 19.10: Site B Phase 2 Settings

Advanced Configuration	
Automatically ping host	10.3.0.1 IP Address

Fig. 19.11: Site B Keep Alive

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.

Description	Local ID	Local IP	Remote ID	Remote IP	Role	Reauth	Algo	Status
ExampleCo London Office	198.51.100.3	198.51.100.3	203.0.113.5	203.0.113.5				Disconnected 

Fig. 19.12: Site A IPsec Status

Failing that, the IPsec logs will 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 [IPsec Troubleshooting](#) section later in this chapter.

Routing and gateway considerations

When the VPN endpoint, in this case a pfSense firewall, is the default gateway for a network there are normally no problems with routing. As a client PC sends traffic, it will go to the pfSense firewall, over the tunnel, and out the other end. However, if the pfSense 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 pfSense firewall is the gateway at Site B, but not Site A, as illustrated in Figure [Site-to-Site IPsec Where pfSense is not the Gateway](#). A client, PC1 at Site B sends a ping to PC2 at Site A. The packet leaves PC1, then through the pfSense firewall at Site B, across the tunnel, out the pfSense firewall at Site A, and on to PC2. But what happens on the way back? The gateway on PC2 is another 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 pfSense firewall. 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 pfSense firewall 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 pfSense firewall the gateway and let it handle the Internet connection instead of the existing gateway.

pfSense-initiated Traffic and IPsec

To access the remote end of IPsec connections from the pfSense firewall itself, “fake” the system out by adding a static route pointing the remote network to the LAN IP address of the pfSense firewall. Note this example presumes the VPN is connecting the LAN interface on both sides. If the IPsec connection is connecting an OPT interface, replace Interface and IP address of the interface accordingly. Because of the way IPsec is tied into the FreeBSD kernel, without the static route the traffic will follow the system routing table, which will likely send this traffic out the WAN interface rather than over the IPsec tunnel. Take Figure [Site-to-Site IPsec](#), for example.

A static route is required on each firewall, which 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](#)). Figure [Site A Static Route to Remote Subnet](#) show the route to be added on each side.

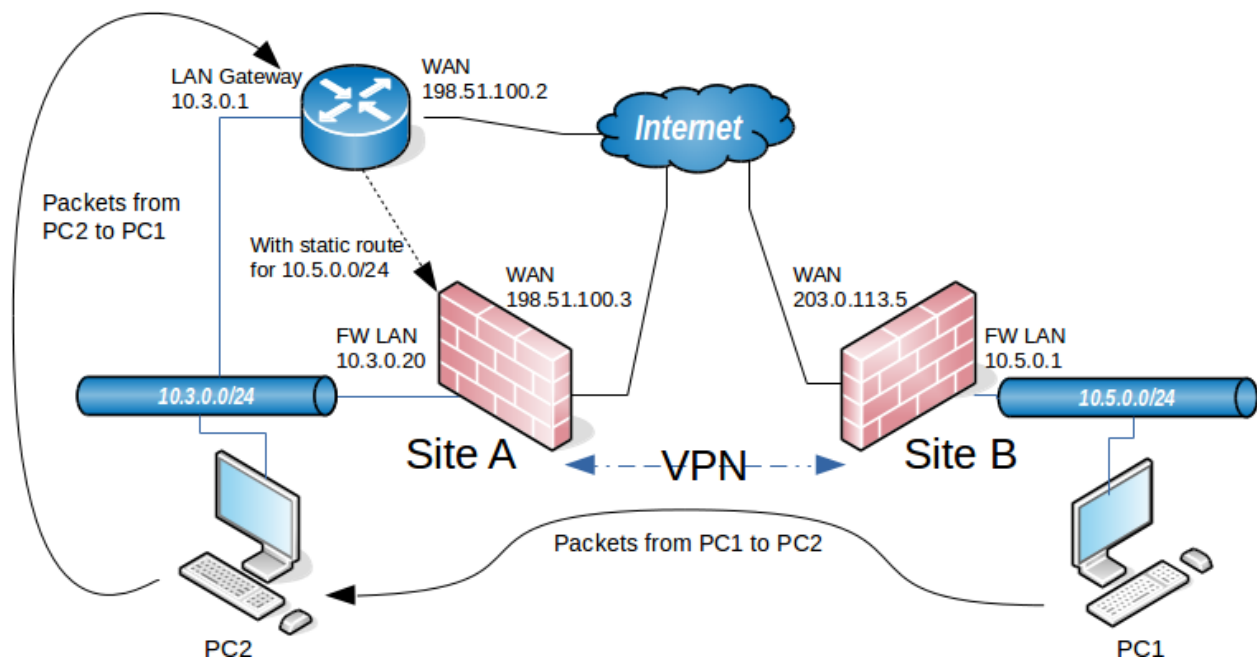


Fig. 19.13: Site-to-Site IPsec Where pfSense is not the Gateway

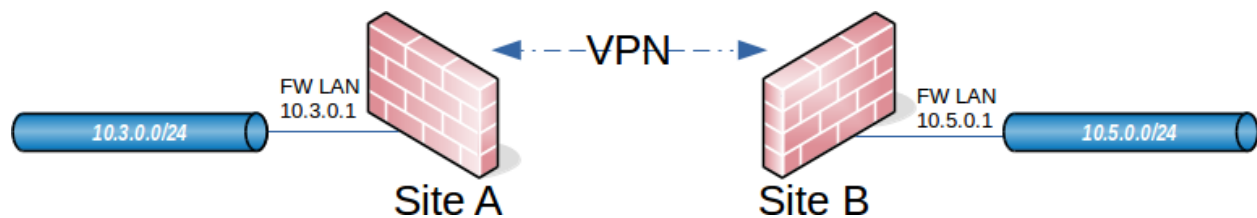


Fig. 19.14: Site-to-Site IPsec

Edit Route Entry	
Destination network	10.5.0.0 / 24 Destination network for this static route
Gateway	IPsecGW - 10.3.0.1 Choose which gateway this route applies to or add a new one first
Disabled	☐ Disable this static route Set this option to disable this static route without removing it from the list.
Description	Route for IPsec connectivity from the firewall You may enter a description here for your reference (not parsed).

Fig. 19.15: Site A Static Route to Remote Subnet

Edit Route Entry	
Destination network	10.3.0.0 / 24 Destination network for this static route
Gateway	IPsecGW - 10.5.0.1 Choose which gateway this route applies to or add a new one first
Disabled	☐ Disable this static route Set this option to disable this static route without removing it from the list.
Description	Route for IPsec connectivity from the firewall You may enter a description here for your reference (not parsed).

Fig. 19.16: Site B Static Route to Remote Subnet

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.

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.

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 be 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.

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.

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.

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.

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.

Example 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

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 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*. If clients will connect by IP address, place the IP address here instead
- Click  **Add** 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
- Click  **Add** to add another 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 addresses on the firewall that clients may use to connect
- Click **Save**

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. 19.17: 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.

The screenshot shows the 'Extended Authentication (Xauth)' configuration page. Under 'User Authentication', a dropdown menu is open showing 'RadAuth' and 'Local Database' as options. Below this, the text 'Source' is visible. Under 'Group Authentication', a dropdown menu shows 'none' as the selected option, with 'Source' written below it.

Fig. 19.18: 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 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.

Client Configuration (mode-cfg)	
Virtual Address Pool	☒ Provide a virtual IP address to clients
10.3.200.0	24
Network configuration for Virtual Address Pool	
Virtual IPv6 Address Pool	☒ Provide a virtual IPv6 address to clients
2001:db8:1:df01::	64
Network configuration for Virtual IPv6 Address Pool	
Network List	☐ Provide a list of accessible networks to clients
Save Xauth Password	☒ Allow clients to save Xauth passwords (Cisco VPN client only). NOTE: With iPhone clients, this does not work when deployed via the iPhone configuration utility, only by manual entry.
DNS Default Domain	☒ Provide a default domain name to clients
example.com	Specify domain as DNS Default Domain
Split DNS	☒ Provide a list of split DNS domain names to clients. Enter a space separated list.
example.com example.org	NOTE: If left blank, and a default domain is set, it will be used for this value.
DNS Servers	☒ Provide a DNS server list to clients
Server #1	10.3.0.1
Server #2	
Server #3	
Server #4	
WINS Servers	☐ Provide a WINS server list to clients
Phase2 PFS Group	☐ Provide the Phase2 PFS group to clients (overrides all mobile phase2 settings)
Login Banner	☐ Provide a login banner to clients

Fig. 19.19: Mobile Clients Pushed Settings

- 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 (Figure *Mobile Clients Phase 1 Creation Prompt*)
- Click the **Tunnels** tab

Support for IPsec Mobile Clients is enabled but a Phase 1 definition was not found.
Please click Create to define one.

[+ Create Phase 1](#)

Fig. 19.20: Mobile Clients Phase 1 Creation Prompt

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

My Certificate Authority Choose the Certificate Authority created earlier

Encryption Algorithm Set to *3DES* (Or *AES 256* if there are no iOS/OS X Devices)

Hash Algorithm Must be set to *SHA1* (Or *SHA256* if there are no iOS/OS X Devices)

DH key group Must be set to *2 (1024 bit)*

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*.

Phase 1 Proposal (Authentication)	
Authentication Method	EAP-MSChapv2 Must match the setting chosen on the remote side.
My identifier	Distinguished name vpn.example.com
Peer identifier	Any This is known as the "group" setting on some VPN client implementations
My Certificate	IKEv2-Server-Cert Select a certificate previously configured in the Certificate Manager.

Phase 1 Proposal (Algorithms)	
Encryption Algorithm	3DES
Hash Algorithm	SHA1 Must match the setting chosen on the remote side.
DH Group	2 (1024 bit) Must match the setting chosen on the remote side.
Lifetime (Seconds)	28800

Fig. 19.21: Mobile Clients Phase 1

- Click **Save**
- Click  **Show Phase 2 Entries** to expand the list of mobile phase 2 entries
- Click  **Add P2** to add a new mobile phase 2.

Figure *Mobile Clients Phase 2* shows the phase 2 options for this mobile tunnel.

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 Must be set to *AES* with *Auto* selected for key length. Also select *3DES* if iOS or OS X Clients will connect.

Hash algorithms Select *SHA1* and *SHA256*

PFS Must be set to *off*

Lifetime Set to 3600

- Click **Save**

General Information	
Disabled	☐ Disable this phase 2 entry without removing it from the list.
Mode	Tunnel IPv4
Local Network	Network 0.0.0.0 / 0 Type Address
NAT/BINAT translation	None / 0 Type Address If NAT/BINAT is required on this network specify the address to be translated
Description	Tunnel Everything A description may be entered here for administrative reference (not parsed).
Phase 2 Proposal (SA/Key Exchange)	
Protocol	ESP ESP is encryption, AH is authentication only.
Encryption Algorithms	☒ AES Auto ☐ AES128-GCM Auto ☐ AES192-GCM Auto ☐ AES256-GCM Auto ☐ Blowfish Auto ☒ 3DES ☐ CAST128 Use 3DES for best compatibility or for a hardware crypto accelerator card. Blowfish is usually the fastest in software encryption.
Hash Algorithms	☐ MD5 ☒ SHA1 ☒ SHA256 ☐ SHA384 ☐ SHA512 ☐ AES-XCBC
PFS key group	off
Lifetime	3600 Seconds

Fig. 19.22: Mobile Clients Phase 2

- Click **Apply changes** (Figure *Apply Mobile Tunnel Settings*) and then the tunnel setup for mobile clients is complete.

The IPsec tunnel configuration has been changed.
The changes must be applied for them to take effect.

✓ Apply Changes

Fig. 19.23: Apply Mobile Tunnel Settings

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*.

Edit Pre-Shared-Secret	
Identifier	jimp@example.com This can be either an IP address, fully qualified domain name or an e-mail address.
Secret type	EAP
Pre-Shared Key	abc123

Fig. 19.24: Mobile IPsec User

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](#).

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.

Windows IKEv2 Client Configuration

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*

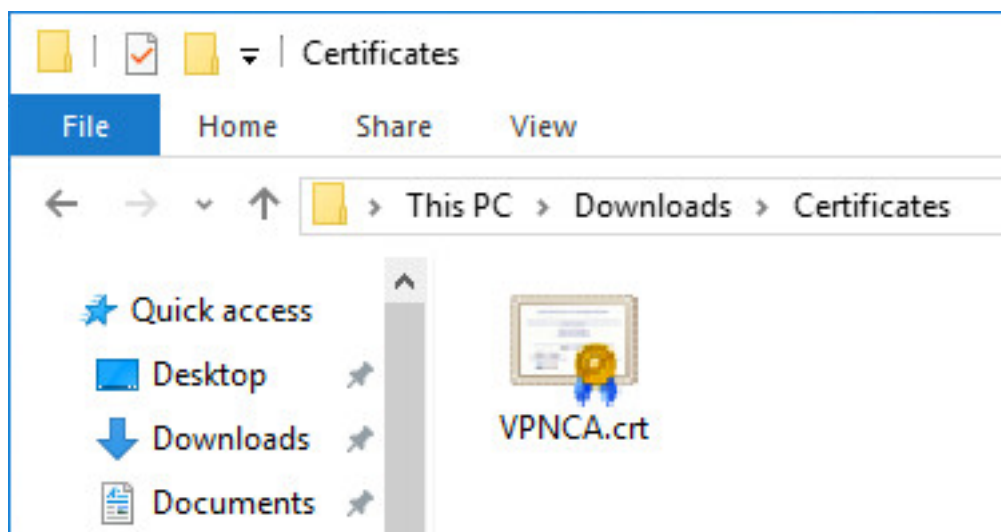


Fig. 19.25: Downloaded CA Certificate

- 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
- 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**

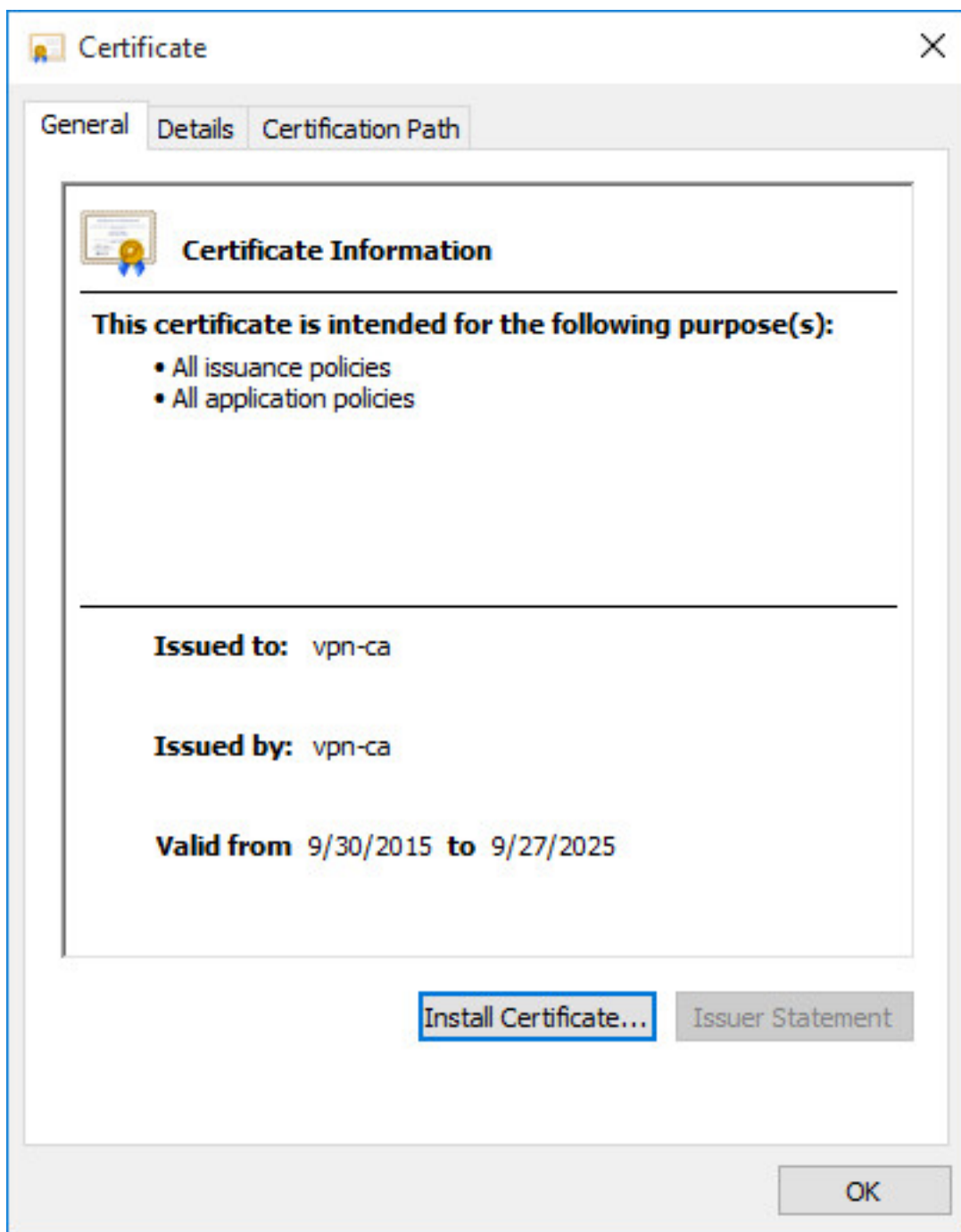


Fig. 19.26: Certificate Properties

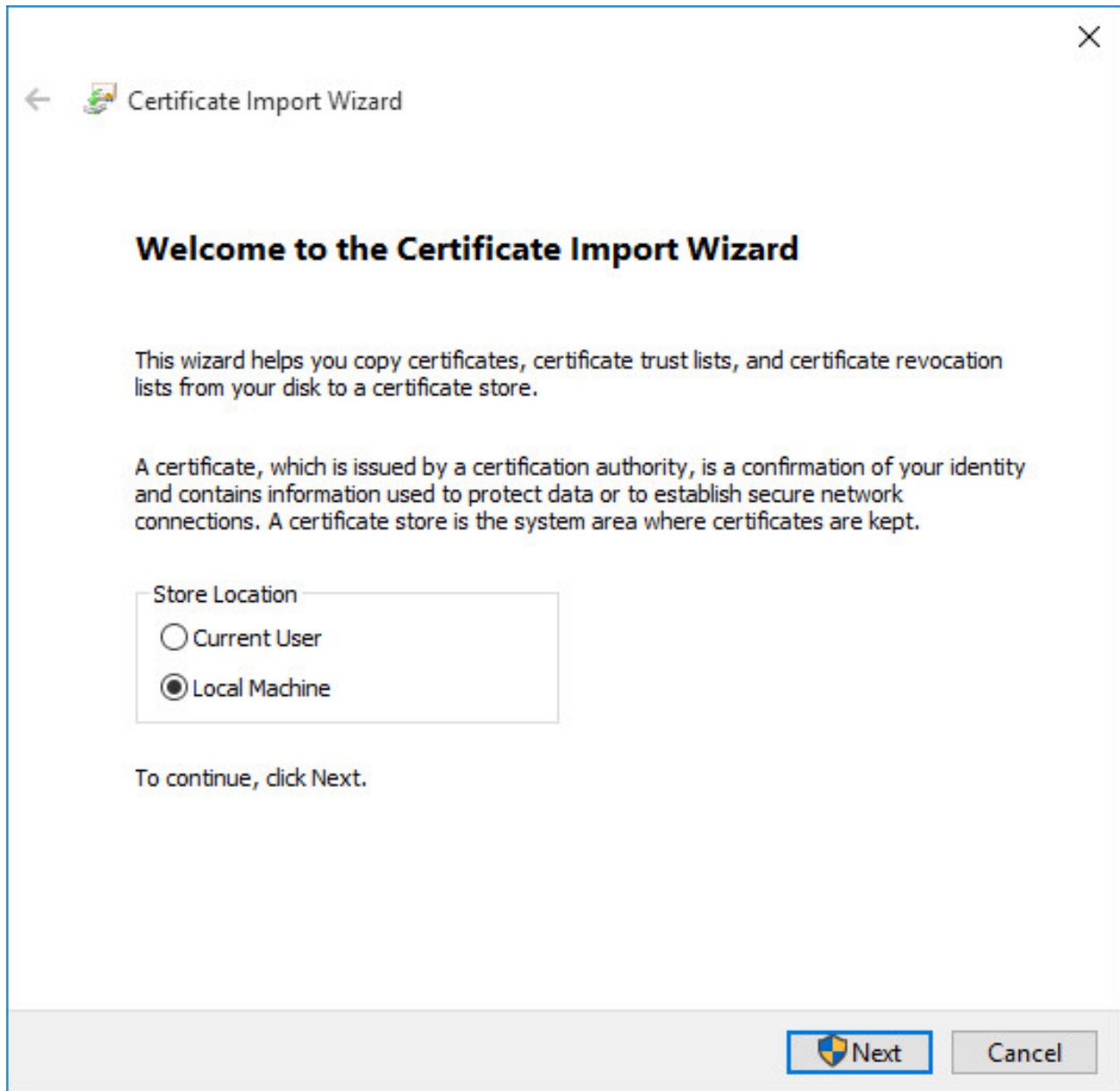


Fig. 19.27: Certificate Import Wizard - Store Location

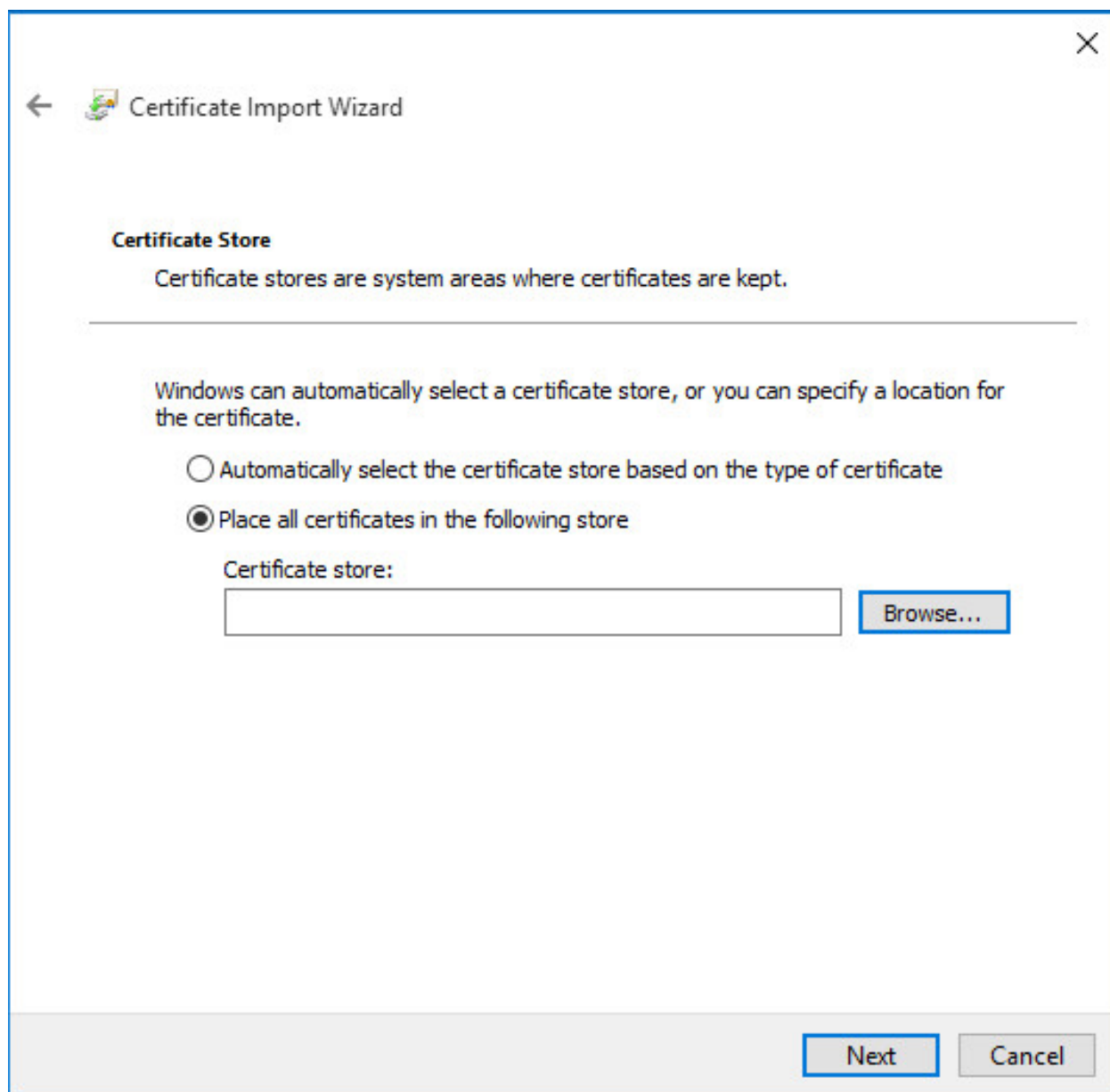


Fig. 19.28: Certificate Import Wizard - Browse for the Store

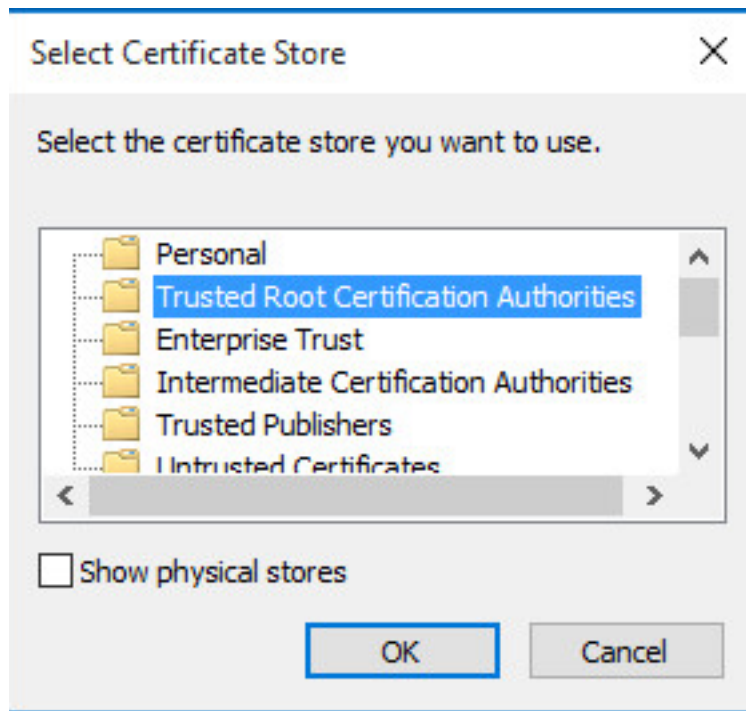


Fig. 19.29: Select Certificate Store

- 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*

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*

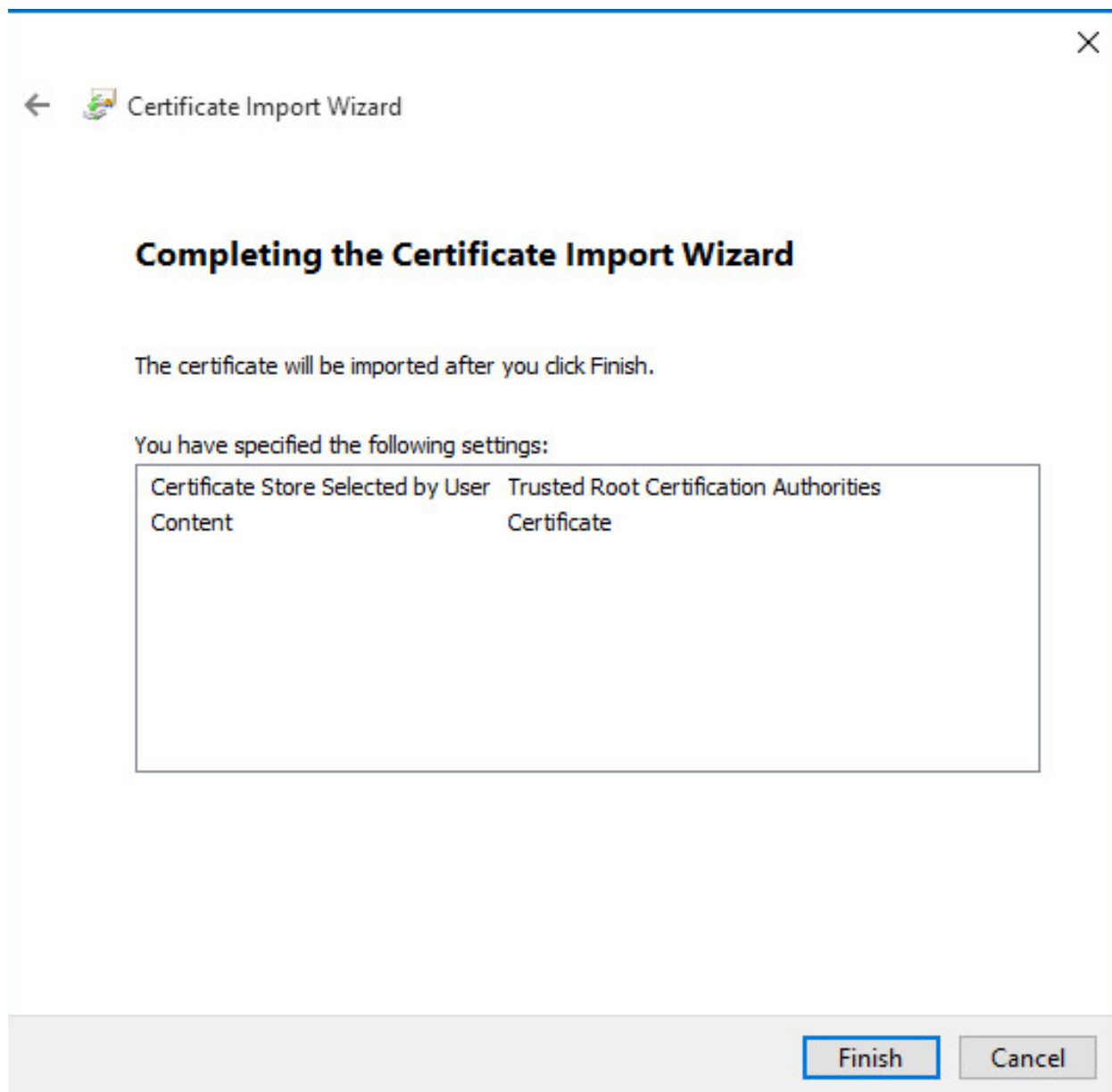


Fig. 19.30: Completing the Certificate Import Wizard

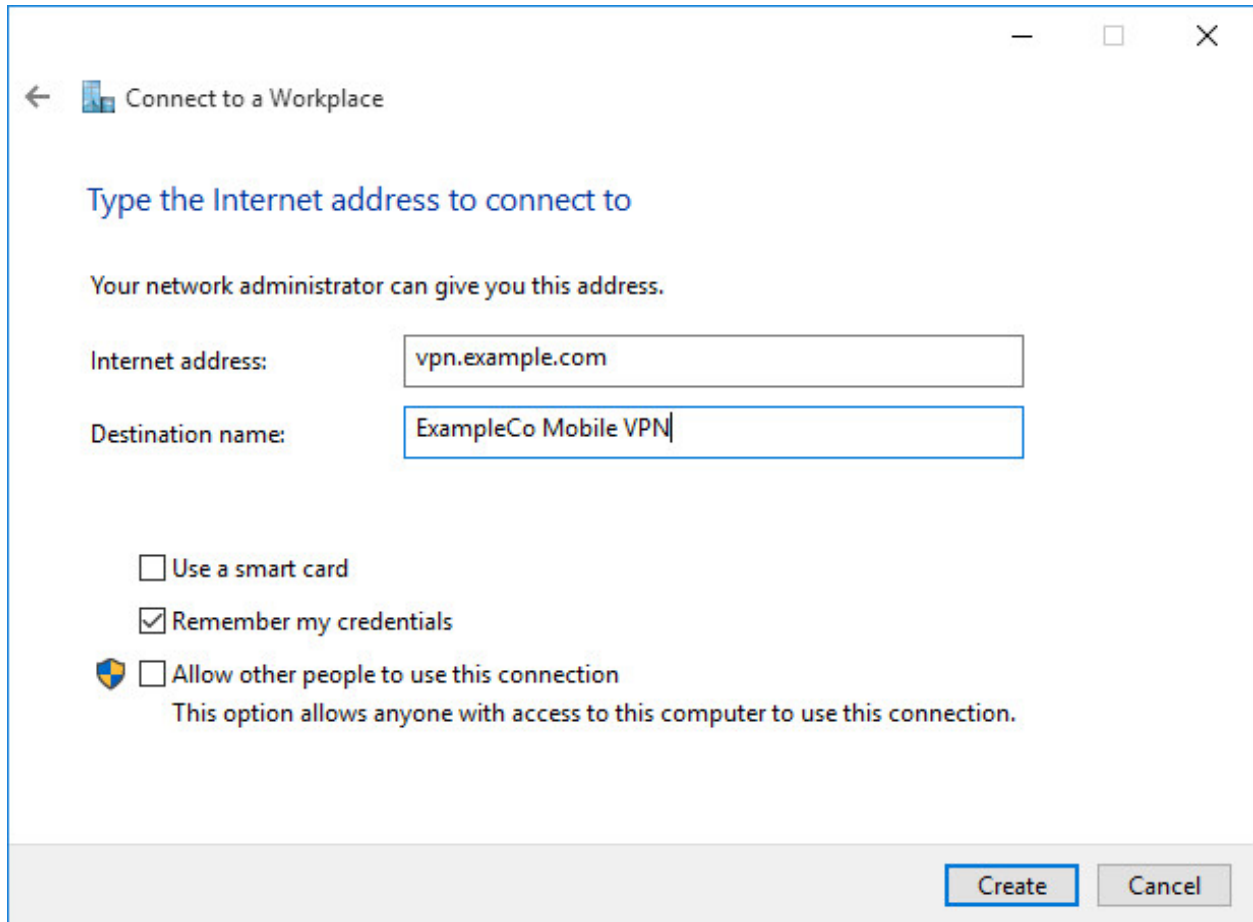


Fig. 19.31: Windows IKEv2 VPN Connection Setup Screen

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.

Disable ECU 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 **DisableIKENNameEkuCheck** and set it to 1.

A reboot may be required to activate the setting.

Ubuntu-based IKEv2 Client Configuration

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*

Note: If the option is not present, double check that **network-manager-strongswan** is installed.

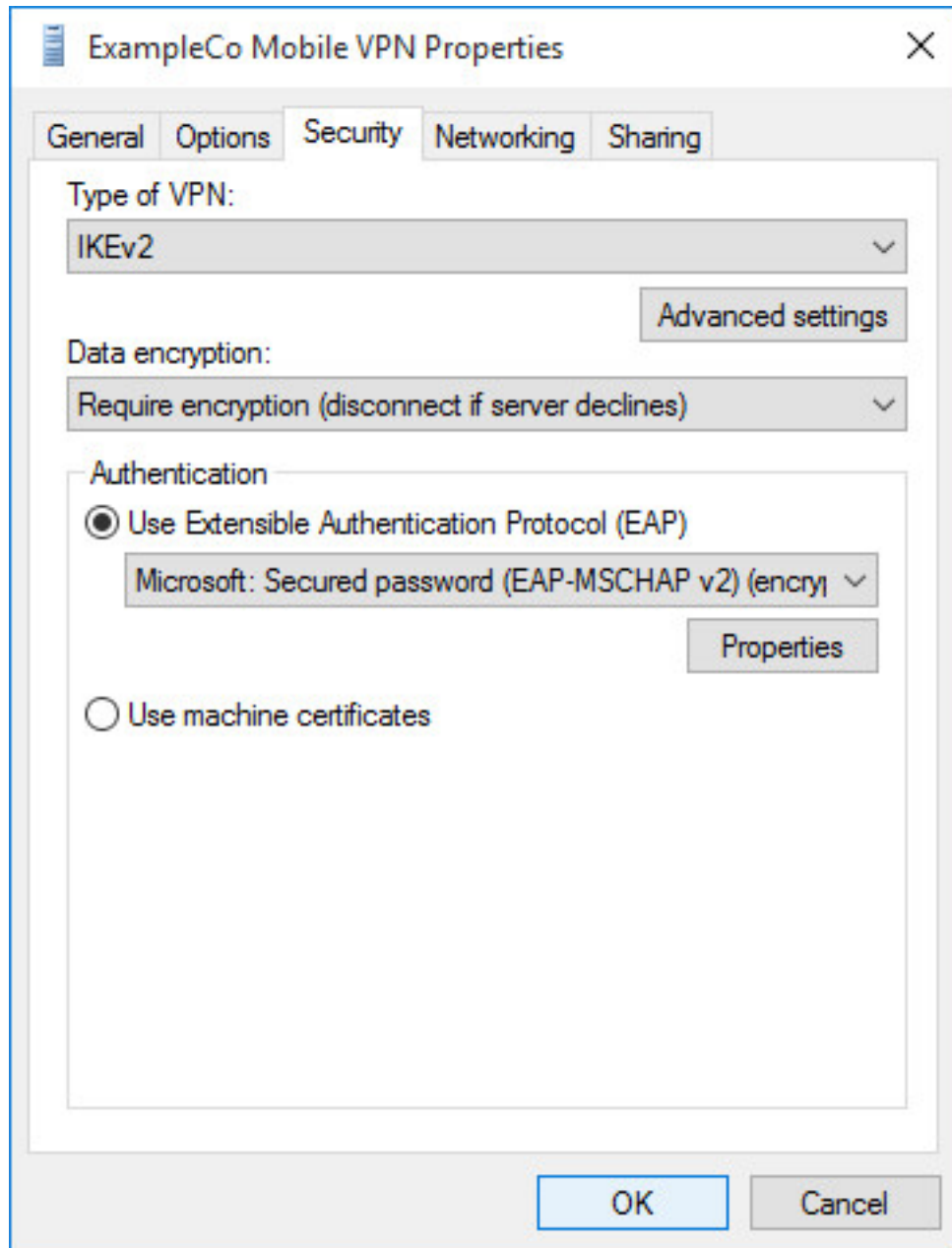


Fig. 19.32: Windows IKEv2 VPN Connection Properties

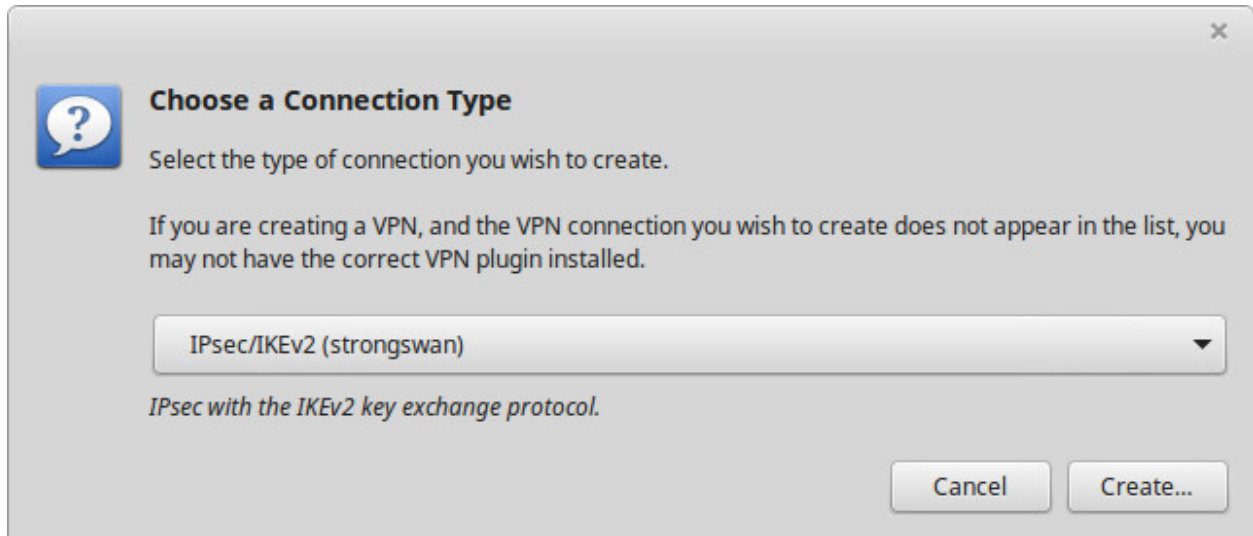


Fig. 19.33: Adding an IKEv2 VPN on Ubuntu

- 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*
- 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

Editing ExampleCo Mobile VPN

Connection name: ExampleCo Mobile VPN

General VPN IPv4 Settings

Gateway

Address: vpn.example.com

Certificate: VPNCA.crt

Client

Authentication: EAP

Username: alice

Options

- ☒ Request an inner IP address
- ☐ Enforce UDP encapsulation
- ☐ Use IP compression

Cancel Save...

Fig. 19.34: Ubuntu VPN Client Settings

Android strongSwan IKEv2 Client Configuration

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:



 Add VPN profile

Profile Name:

ExampleCo Mobile VPN

Gateway:

vpn.example.com

Type:

IKEv2 EAP (Username/Password)

Username:

alice

Password:

• • • • •

CA certificate:

☒ Select automatically

☐ Show advanced settings

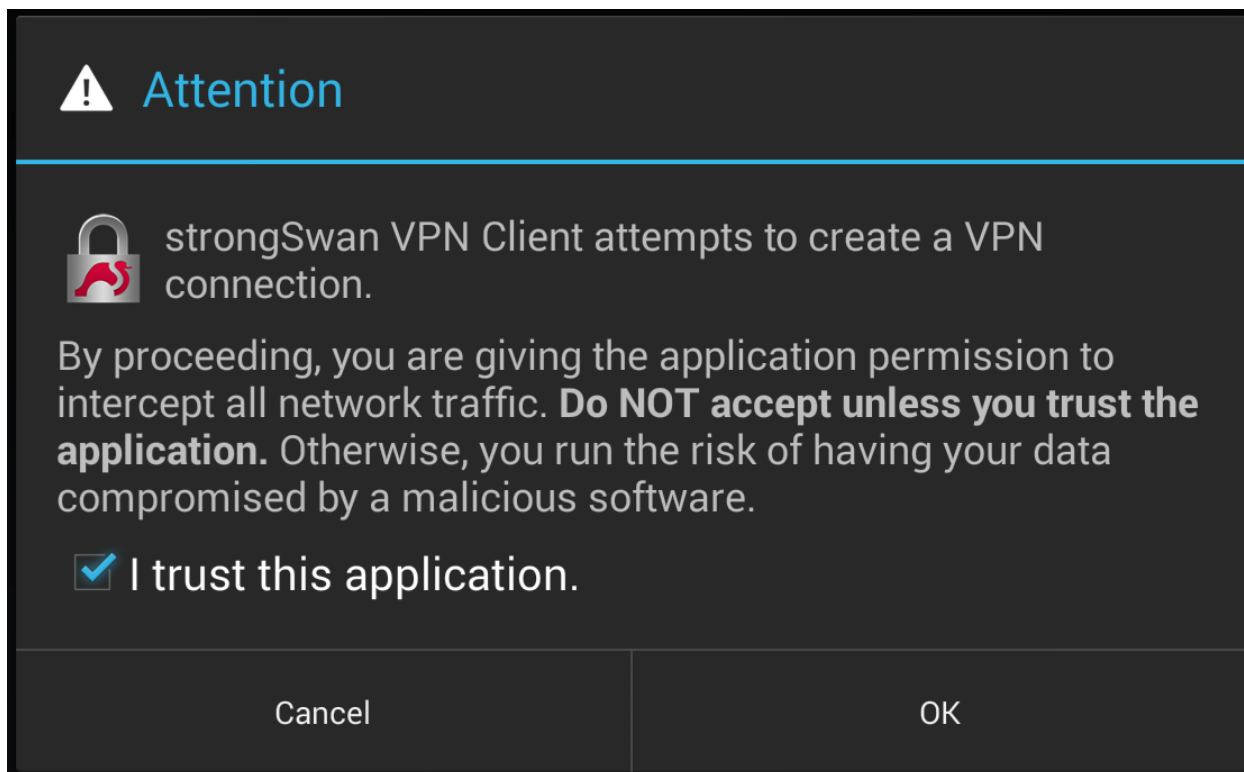


Fig. 19.36: Android strongSwan Client Settings

- Open the strongSwan app
- Tap **Disconnect** on the desired VPN

OS X IKEv2 Client Configuration

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**
- Locate the imported certificate under **System, Certificates** as shown in Figure *OS X Keychain Access System Certificate List*
- Click the Certificate

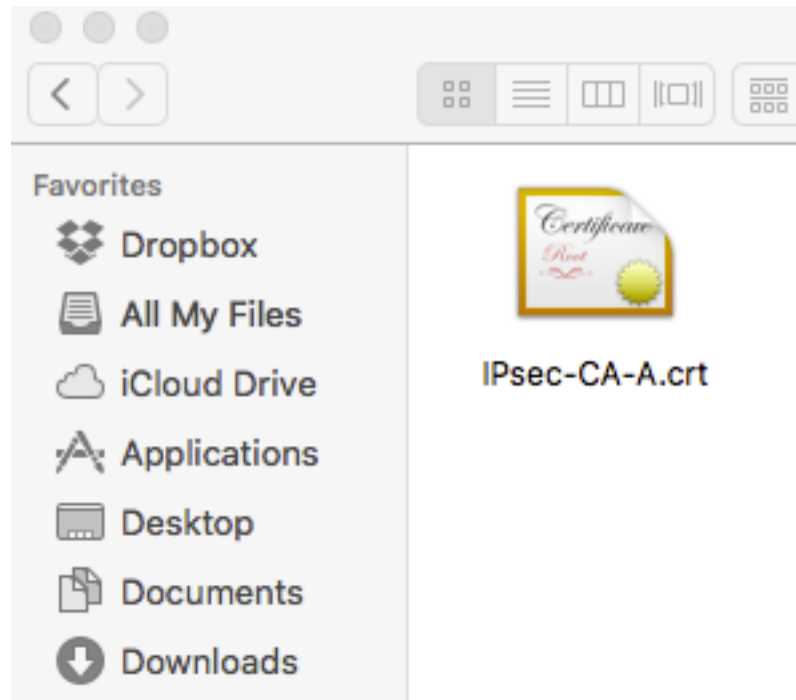


Fig. 19.37: OS X Certificate File in Finder

- 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.

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*
- 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**

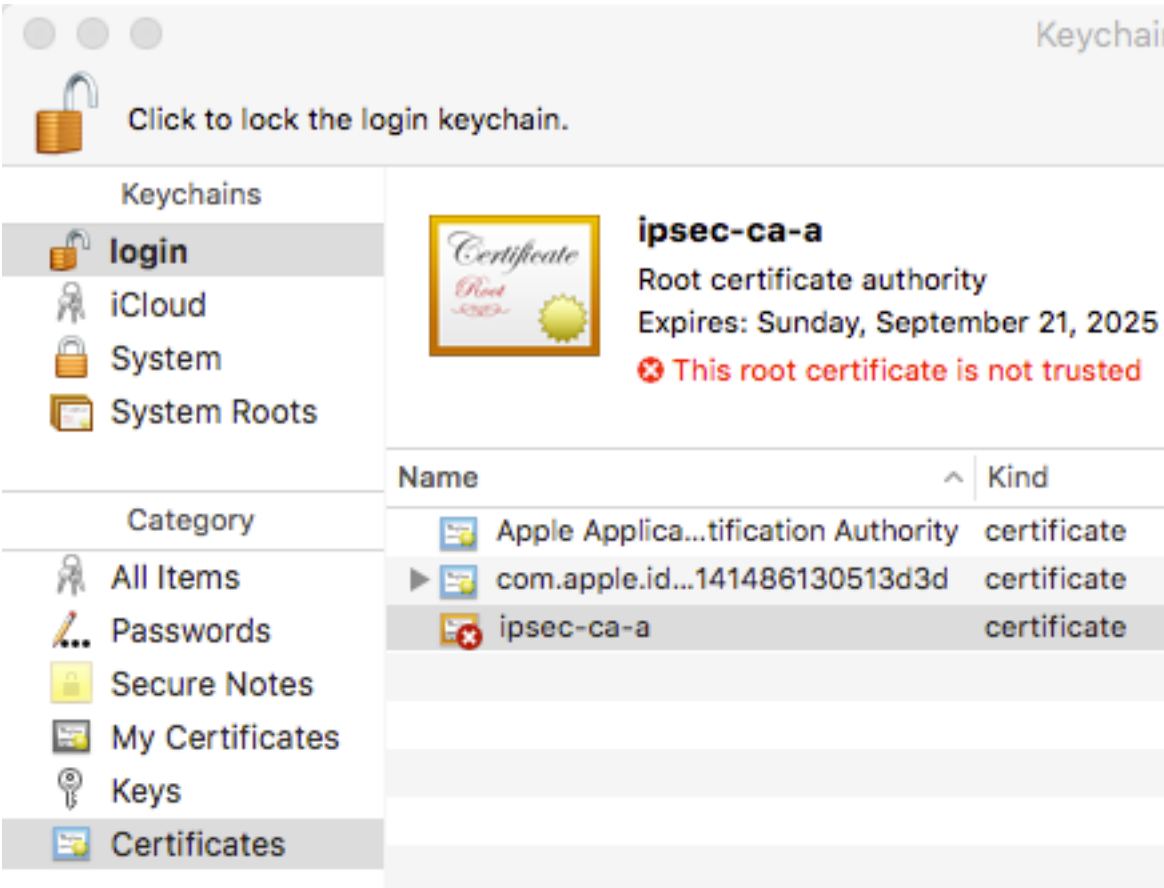


Fig. 19.38: OS X Keychain Access Login Certificate List

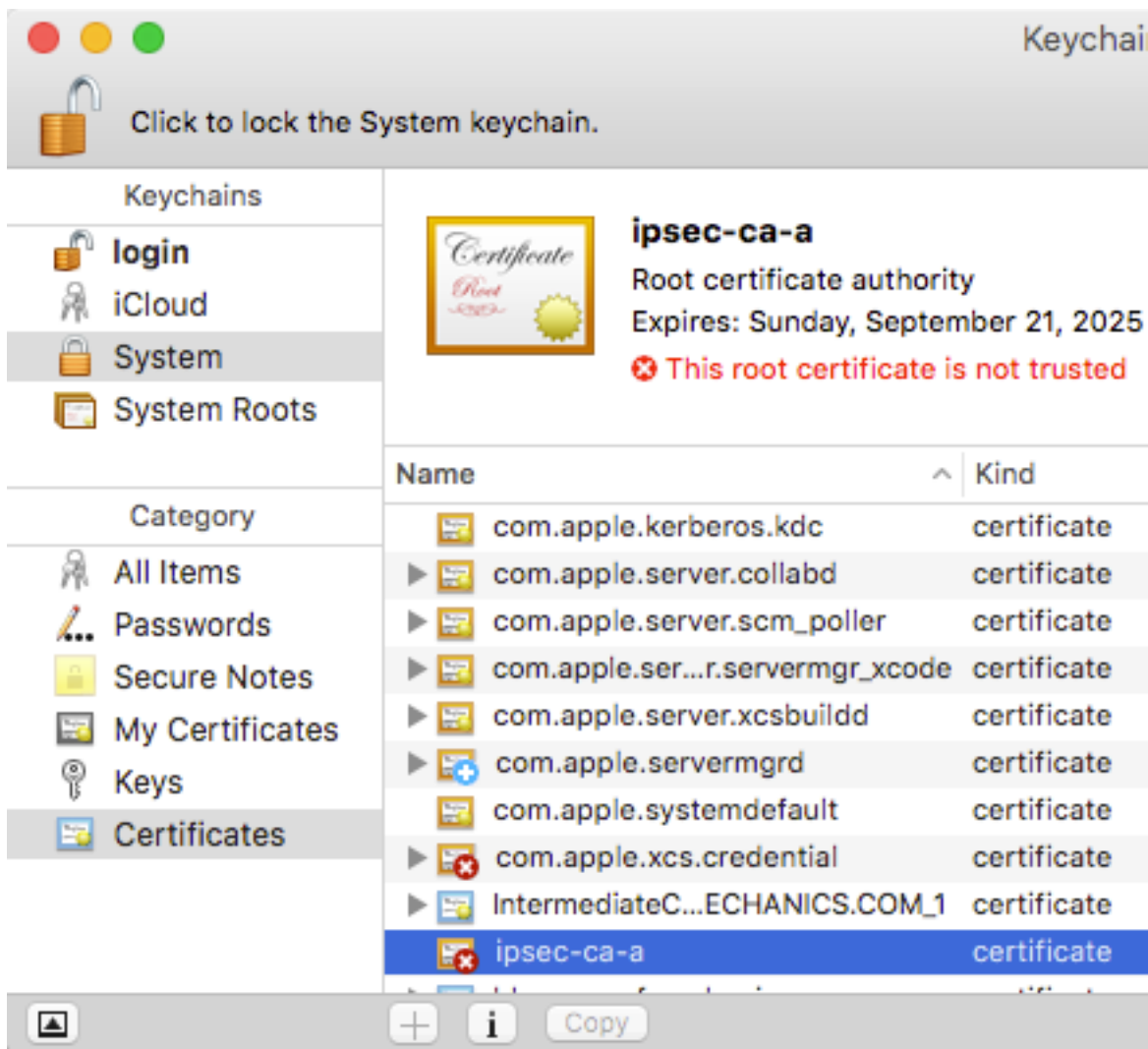


Fig. 19.39: OS X Keychain Access System Certificate List



Fig. 19.40: OS X Certificate Trust Settings



Fig. 19.41: OS X Add Network Button

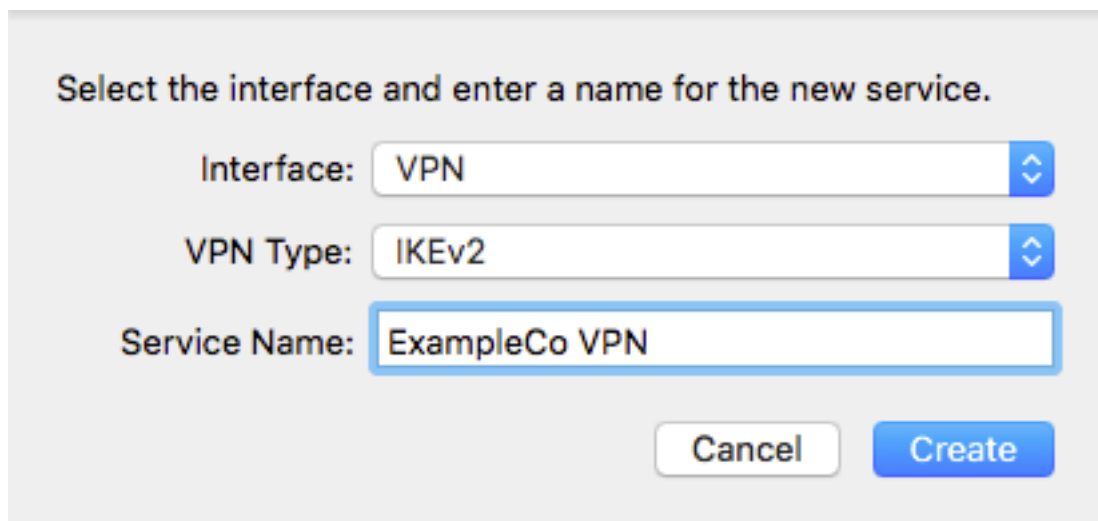


Fig. 19.42: OS X Create VPN Prompt

- 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*

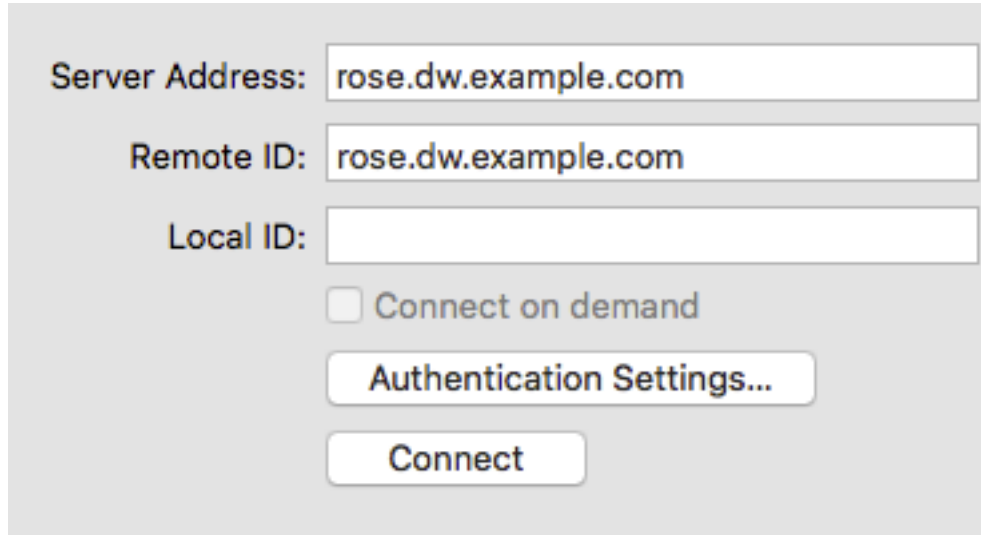
The screenshot shows the 'OS X IKEv2 VPN Settings' dialog box. It has three text input fields: 'Server Address' with the value 'rose.dw.example.com', 'Remote ID' with the value 'rose.dw.example.com', and 'Local ID' which is empty. Below these fields is an unchecked checkbox labeled 'Connect on demand'. At the bottom are two buttons: 'Authentication Settings...' and 'Connect'.

Fig. 19.43: OS X IKEv2 VPN Settings

- Click **Authentication Settings**
- Select **Username**
- 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.

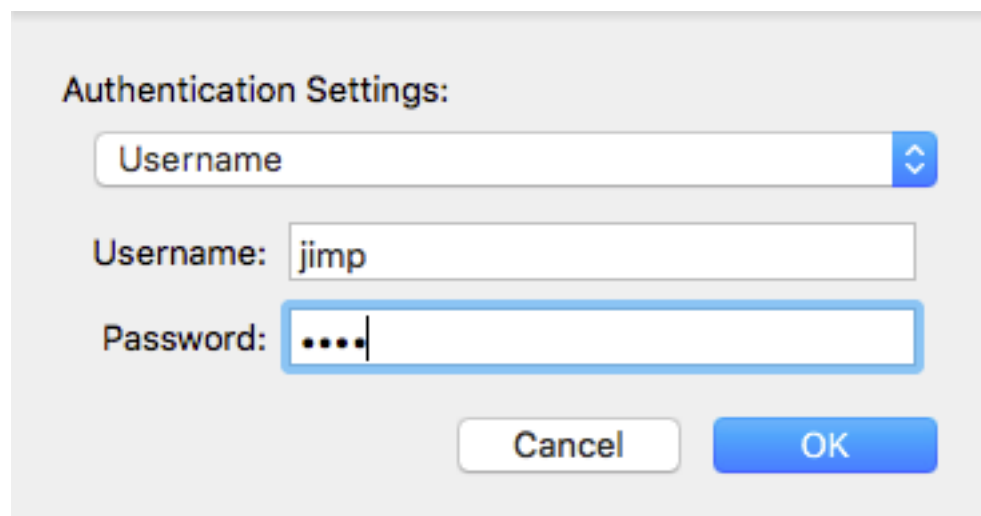
The screenshot shows the 'OS X IKEv2 VPN Authentication Settings' dialog box. It has a dropdown menu labeled 'Username' with a blue arrow icon on the right. Below it are two text input fields: 'Username' with the value 'jimp' and 'Password' with four dots. At the bottom are two buttons: 'Cancel' and 'OK'.

Fig. 19.44: 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*.

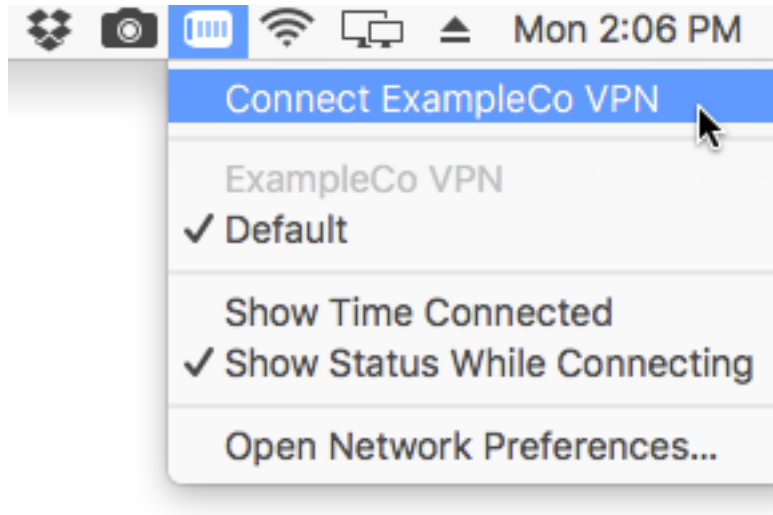


Fig. 19.45: OS X VPN Status Menu

iOS 9 IKEv2 Client Configuration

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*

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*
- 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.

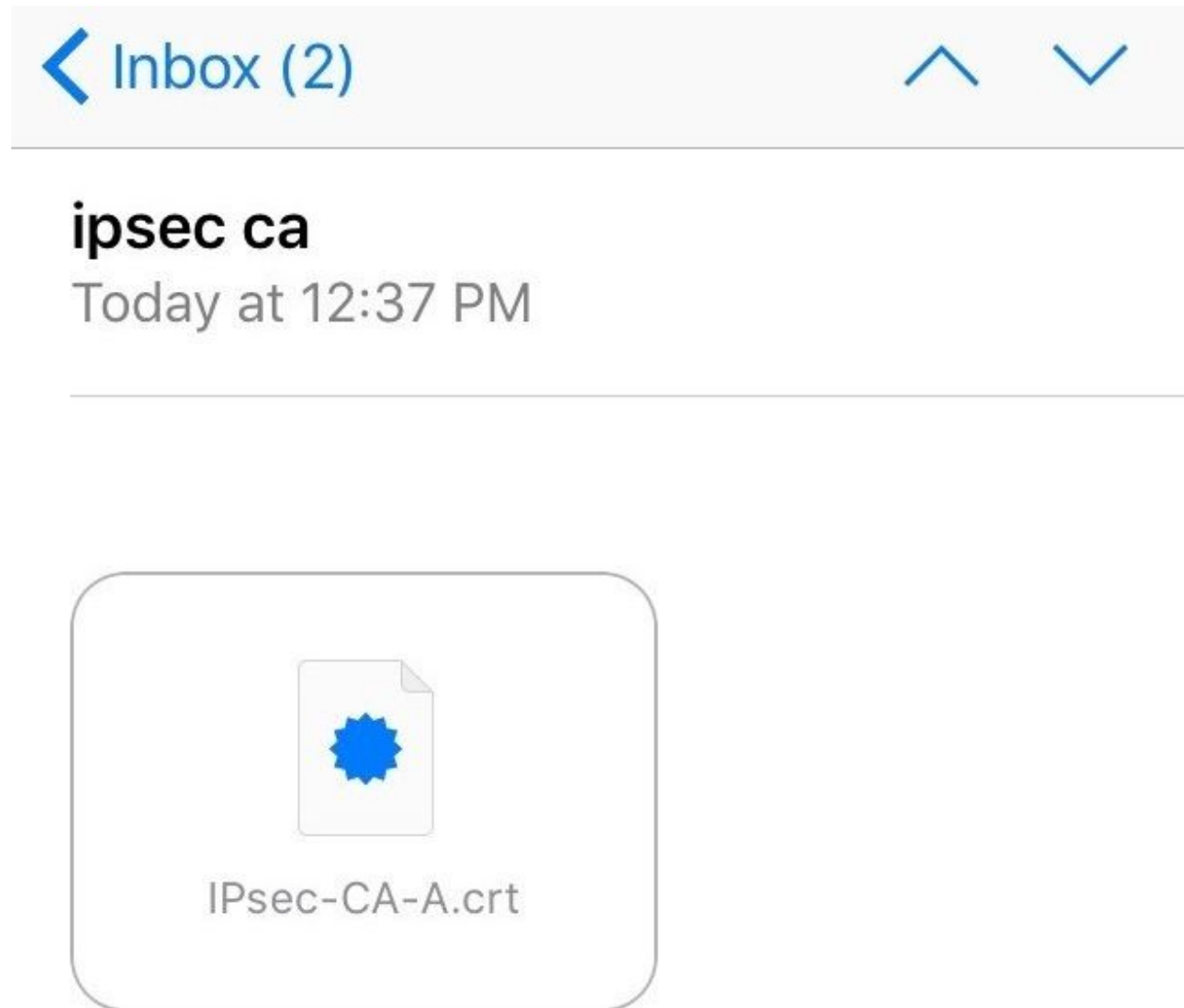


Fig. 19.46: iOS Mail Client Receiving CA Certificate

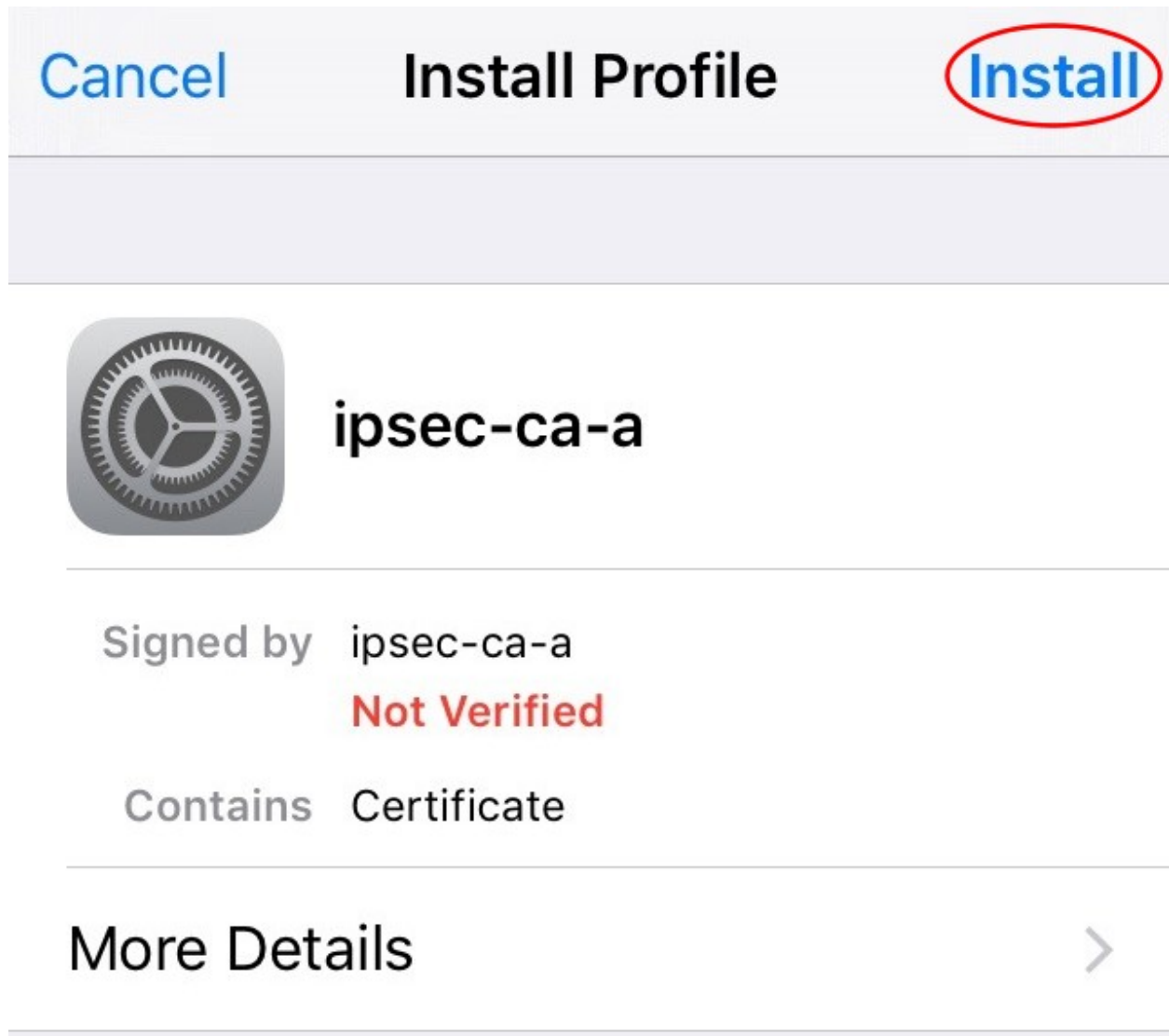


Fig. 19.47: iOS CA Certificate Install Profile Prompt

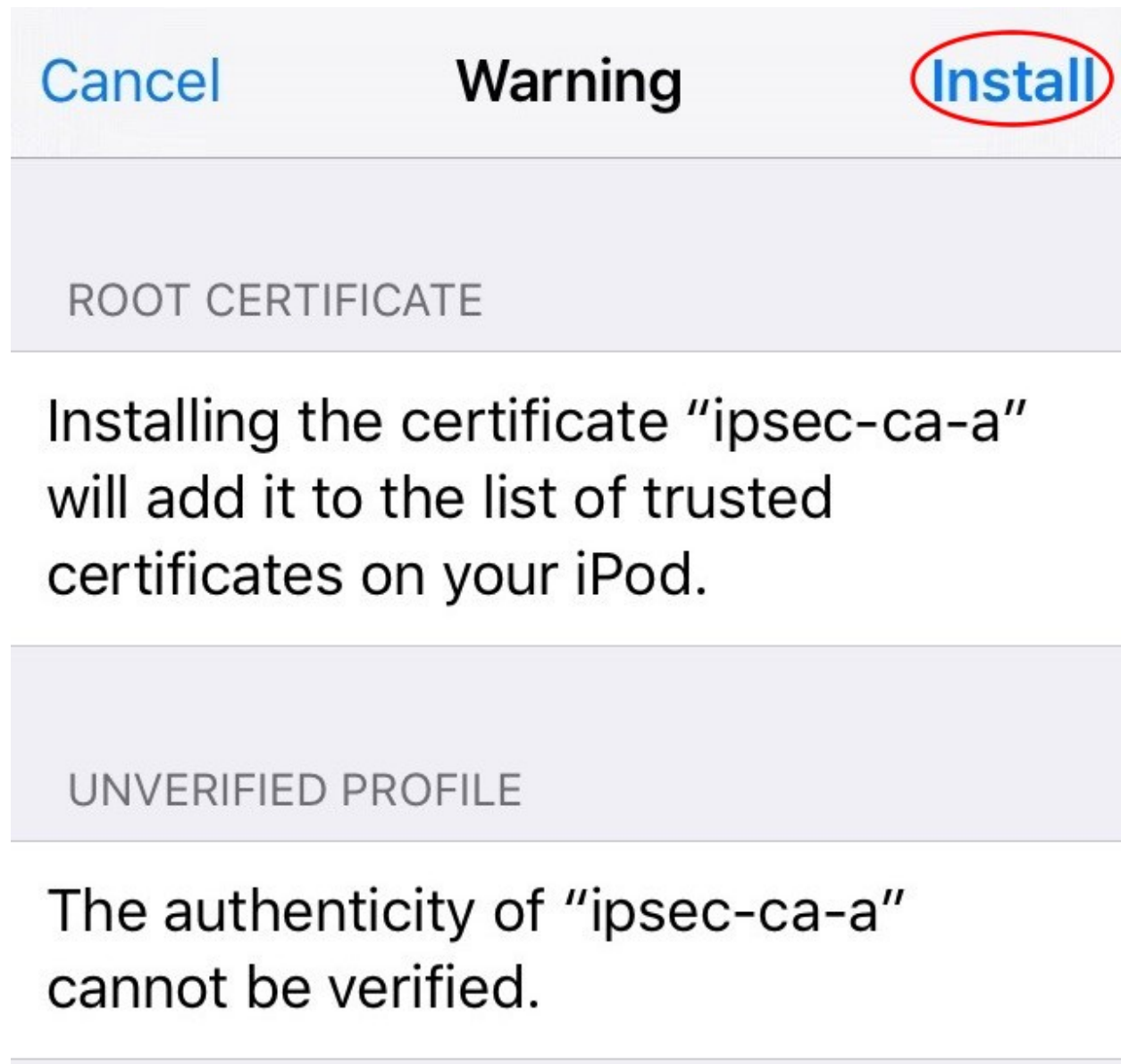


Fig. 19.48: iOS CA Certificate Install Warning

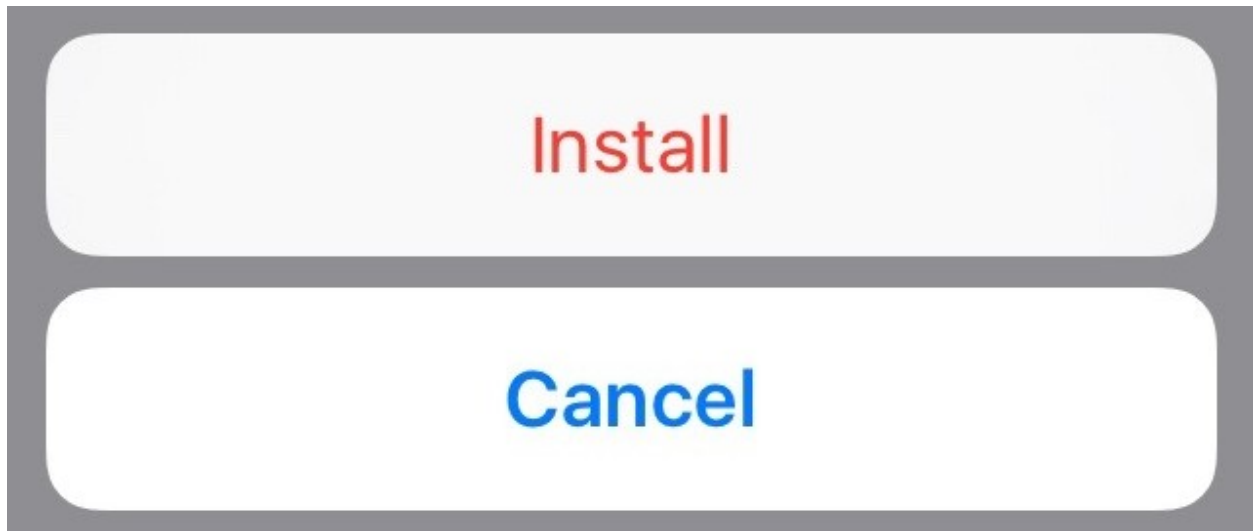


Fig. 19.49: iOS CA Certificate Confirmation Prompt

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*

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:

iPod 

11:16 AM

Cancel

ExampleCo VPN

Done

TypeIKEv2

DescriptionExampleCo VPN

Serverrose.dw.example.com

Remote IDrose.dw.example.com

Local ID

AUTHENTICATION

User AuthenticationUsername >

Usernamejimp

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.

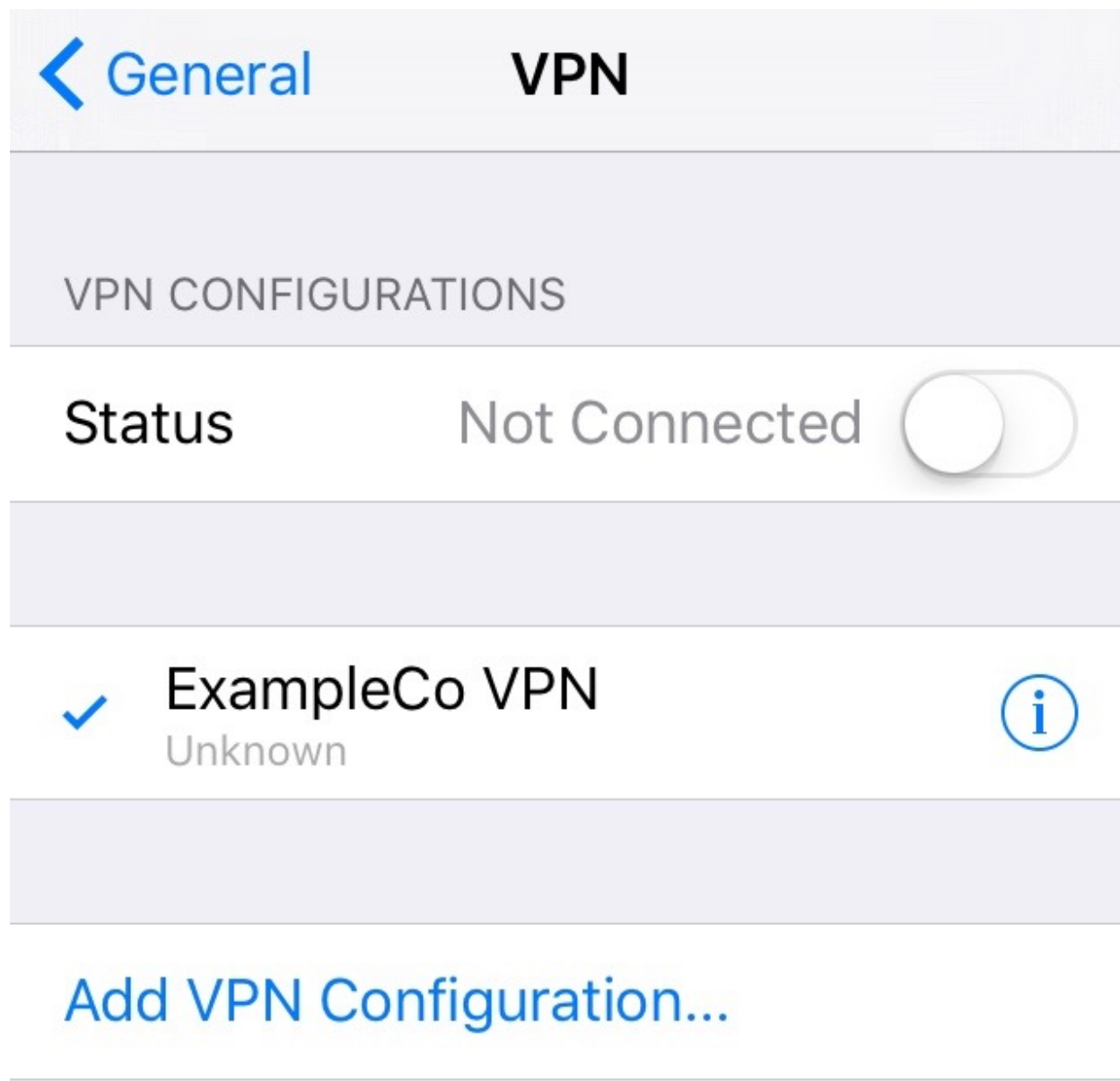


Fig. 19.51: iOS VPN List

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.

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 *pfSense-initiated Traffic and IPsec*, 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 [IPsec Troubleshooting](#) 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 [IPsec Troubleshooting](#) section.

IPsec Troubleshooting

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.

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 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.

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.

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 [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 [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 [Routing and gateway considerations](#)), an incorrectly specified remote subnet on the tunnel definition, or to a tunnel which has been disabled.

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 [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 [Using tcpdump from 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.

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.

“Random” Tunnel Disconnects/DPD Failures on Embedded Routers

If IPsec tunnels are dropped on an ALIX or other embedded 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.

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.

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.

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
```

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
```

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 proposals 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
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
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.

Configuring Third Party IPsec Devices

Any VPN device which supports standard IPsec may be connected to a device running pfSense. 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 [IPsec Troubleshooting](#). 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.

Lifeytes 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.

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 [Site-to-Site](#) 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 3deshal 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 3deshal
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 3deshal 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
crypto map outside_map 20 set peer 198.51.100.3
crypto map outside_map 20 set transform-set 3deshal
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
```

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.

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.

OPENVPN

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.

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 server certificate for this OpenVPN server instance here. 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 [Site-to-Site Example Configuration \(SSL/TLS\)](#) 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 *Bridged OpenVPN Connections*

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 servers 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 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.

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.

Server Hostname Resolution

When **Infinitely Resolve Server** is checked, the server host name will be resolved on each connection attempt. When unchecked, OpenVPN will only attempt to resolve it once. When using a hostname for the remote server address, this option should be checked.

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.

Using the OpenVPN Server Wizard for Remote Access

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.

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.

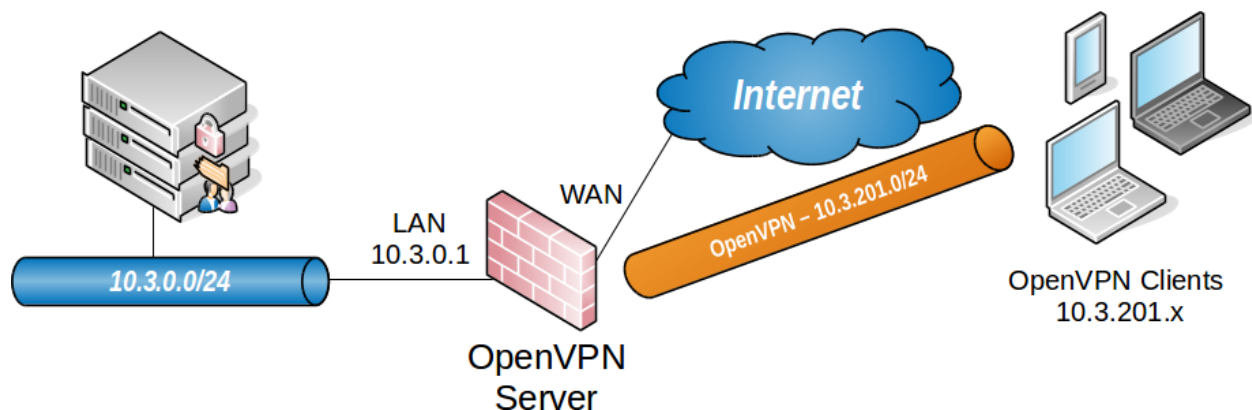


Fig. 20.1: OpenVPN Example Remote Access Network

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

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.

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.

Configuring 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](#).

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.

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 book. 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*

OpenVPN Client Installation

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.

Automatic 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.

Automatic 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.

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.

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 *Android 4.x and later*.

OpenVPN Connect (iOS/Android) Used with the OpenVPN Connect client on iOS or Android described in *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 [Windows Installation](#) 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.

Windows Installation

The OpenVPN project provides an installer for Windows 2000 through Windows 10, downloadable from [The OpenVPN Community Downloads Page](#). At the time of this writing, the best version for most Windows users is 2.3.x-I60x installer. The 2.3 series is the most current stable release

The installation is straightforward, accept all the defaults. The installation will create a new **Local Area Connection** on the client system for the *tun* interface. 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.

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.

Mac OS X Clients and Installation

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 \$9 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](#)
- 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
- 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.
- Click on it and then click **Details** as shown in Figure [Viscosity Menu](#) to see connection information

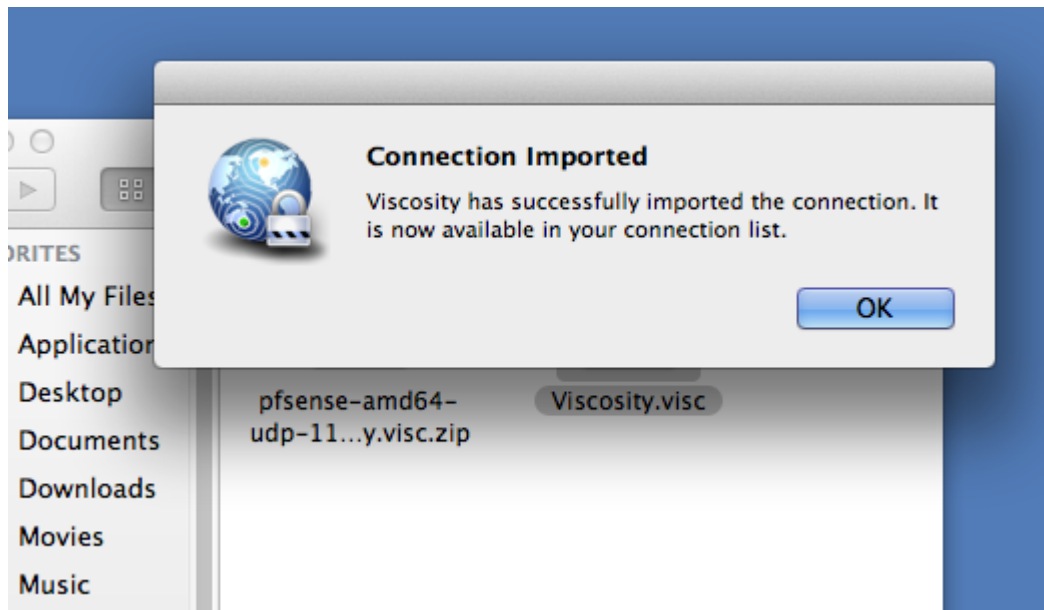


Fig. 20.2: Viscosity Import

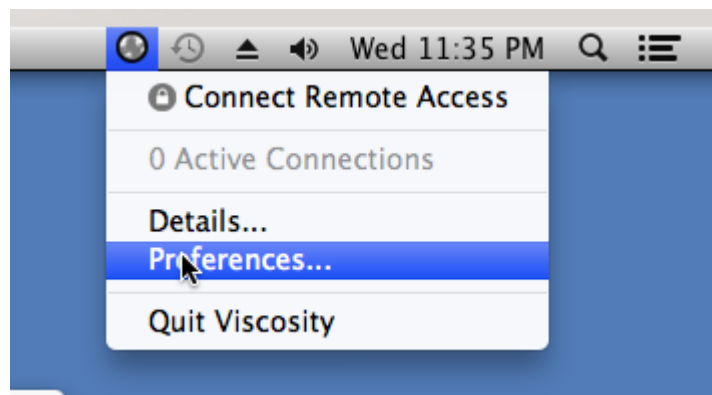


Fig. 20.3: Viscosity Preferences

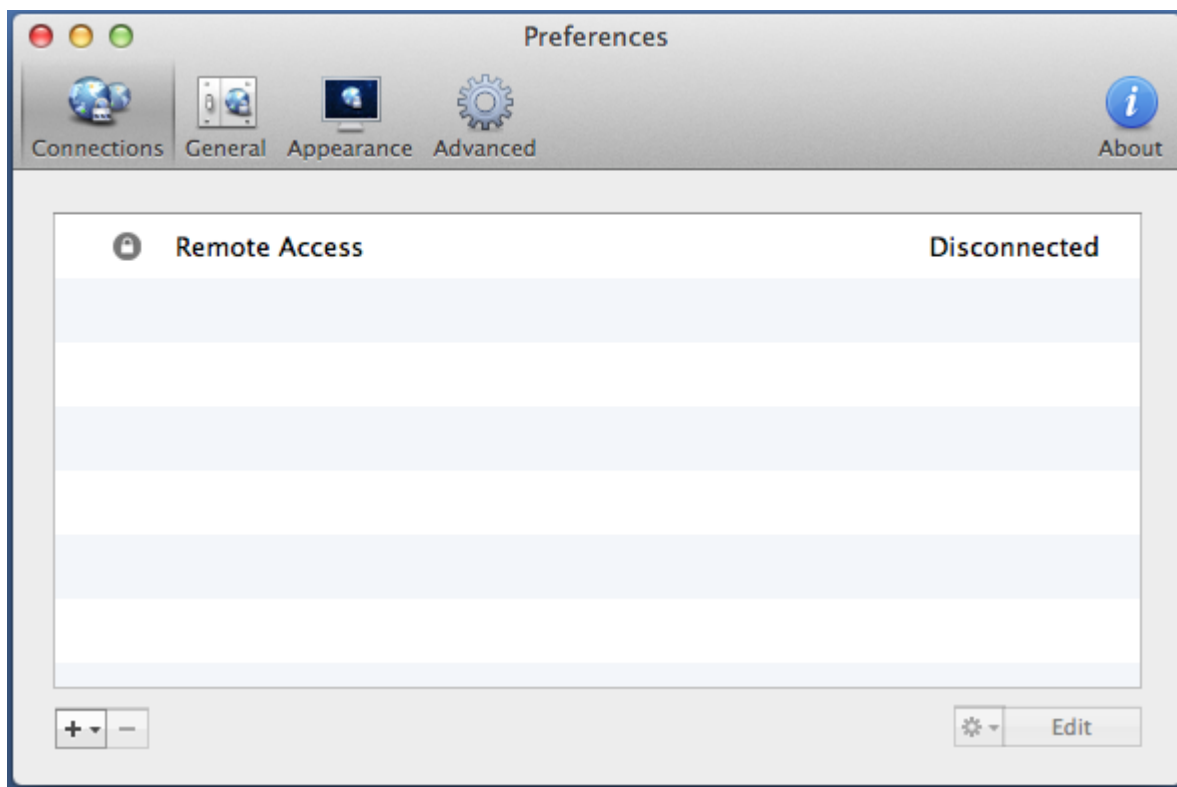


Fig. 20.4: Viscosity View Connections

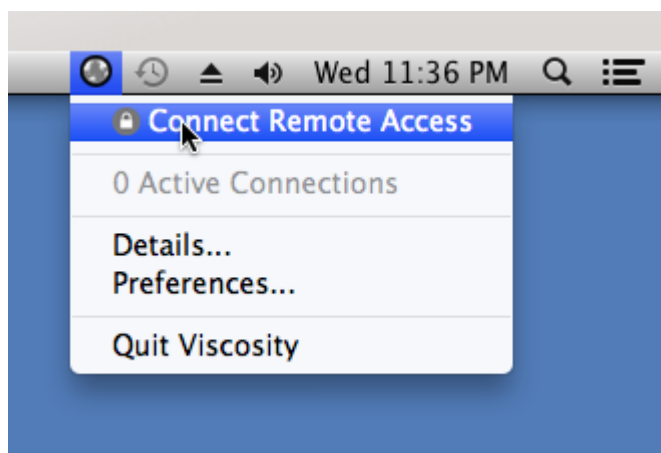


Fig. 20.5: Viscosity Connect

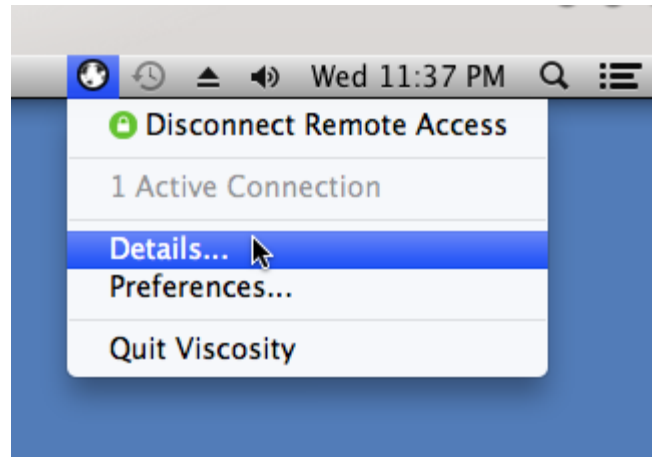


Fig. 20.6: Viscosity Menu

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](#).

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:

- 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

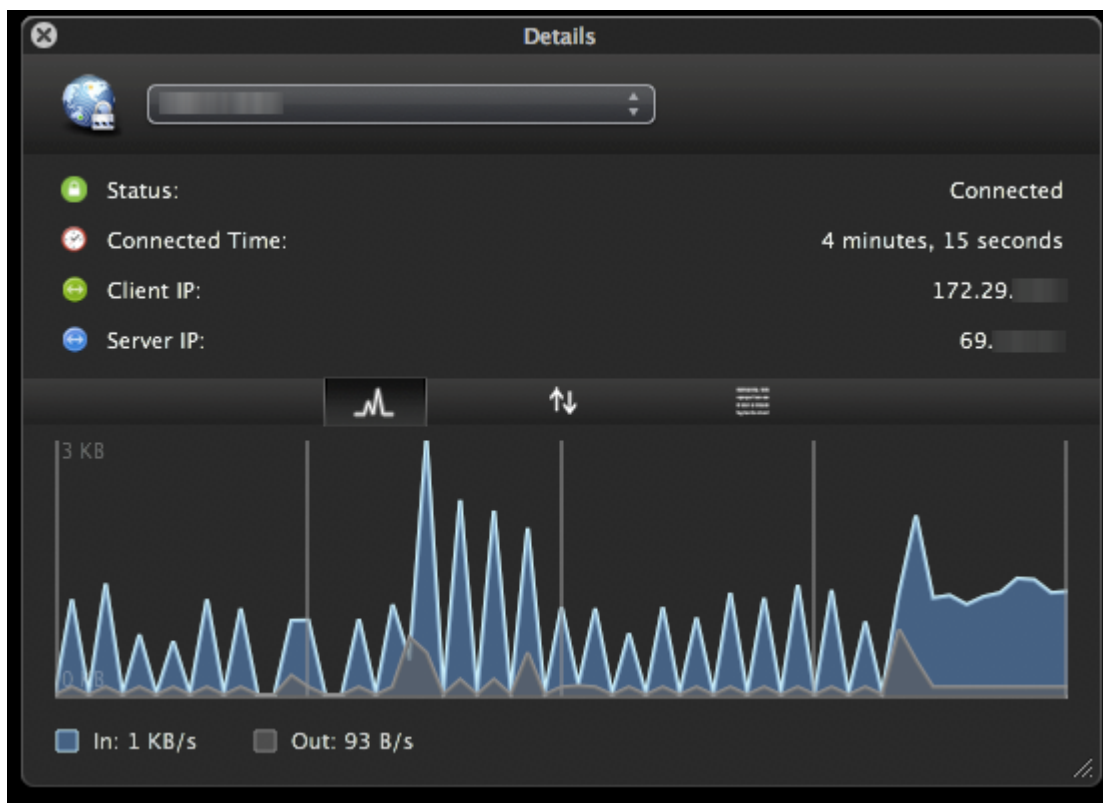


Fig. 20.7: Viscosity Details

- 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

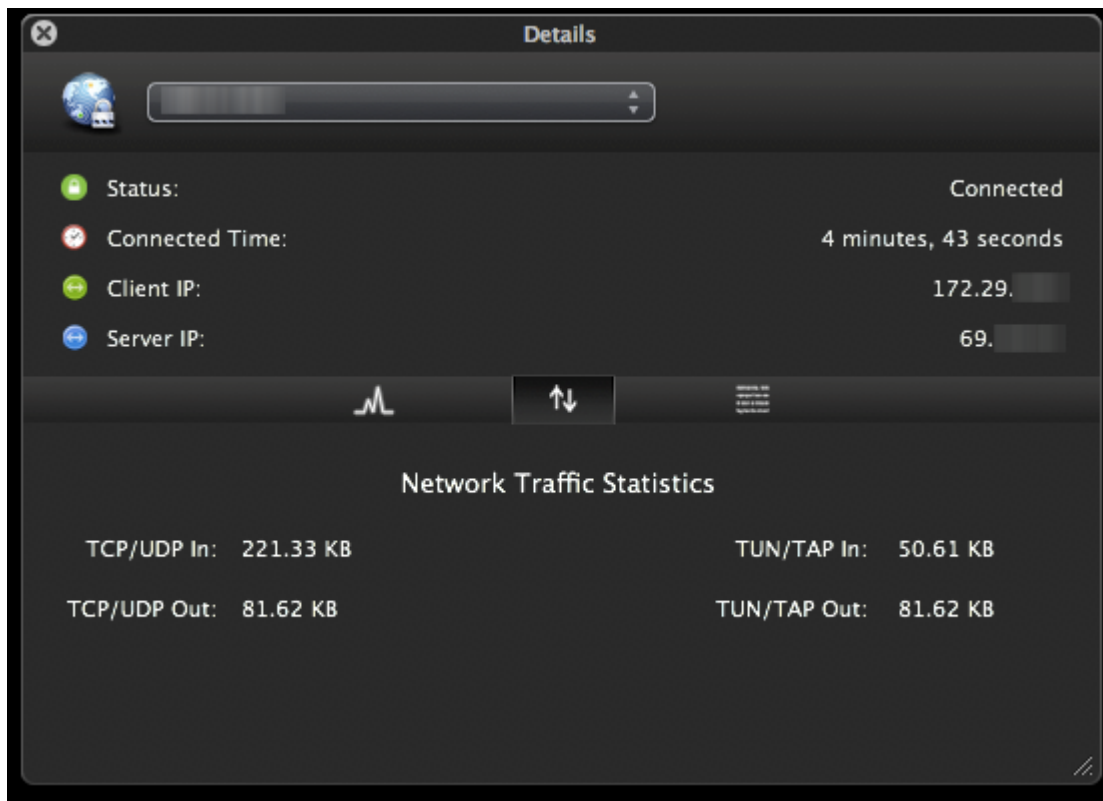


Fig. 20.8: Viscosity Details: Traffic Statistics

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.

Android 4.x and later

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.

The 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)

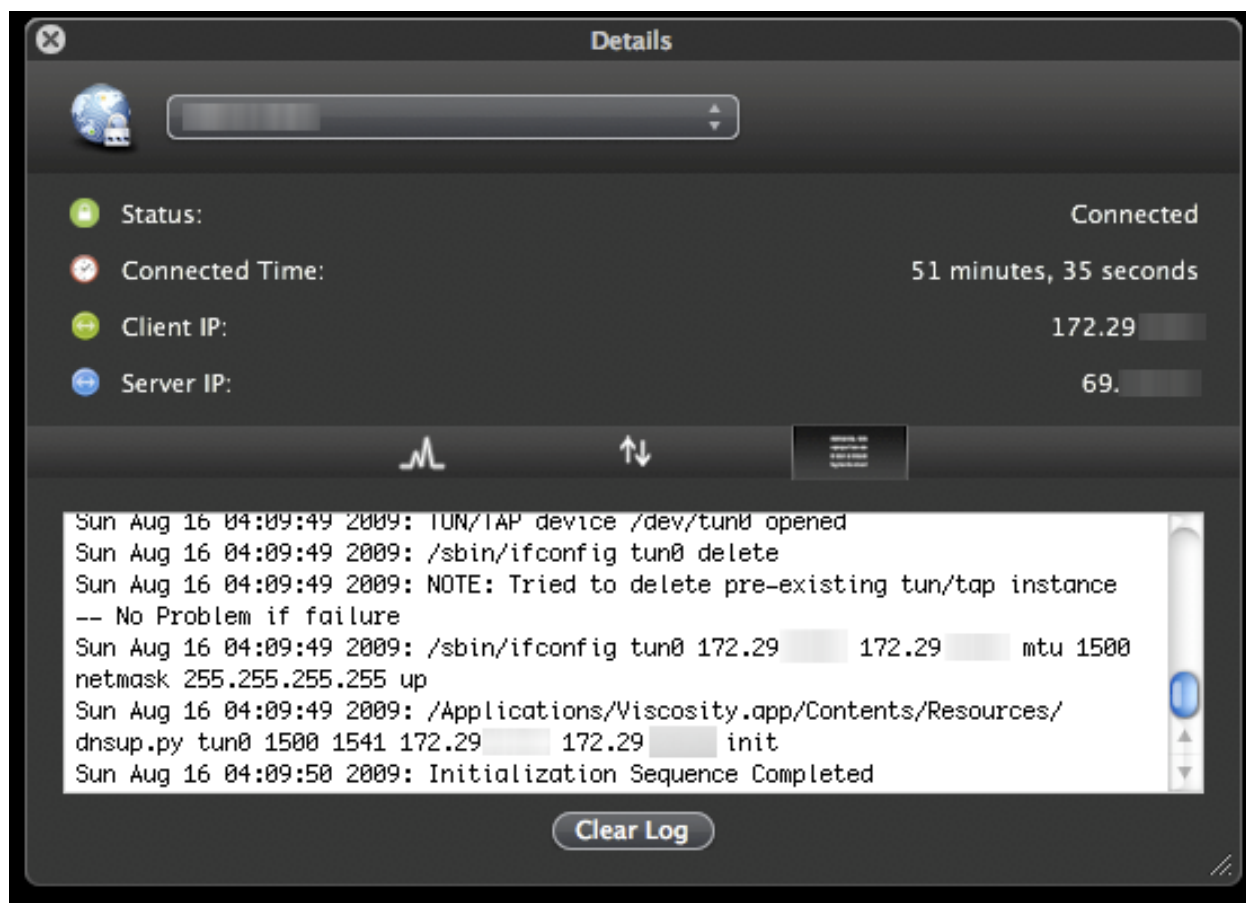


Fig. 20.9: Viscosity Details: Logs

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.

FreeBSD Installation

If the client has a stock FreeBSD installation, OpenVPN may be found in the ports collection.

To install OpenVPN, run the following as **root**:

```
# cd /usr/ports/security/openvpn && make install clean
```

Alternately, it can be installed from packages:

```
# pkg install openvpn
```

Linux Installation

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 book won't cover any 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.

Manual Client Configuration

Performing a manual client installation instead of using the OpenVPN Client Export Package requires more steps to install the software and settings onto the client PCs. 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.

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.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to watch the September and October 2015 Hangouts on Remote Access VPNs which covers client installations for most operating systems.

Site-to-Site Example (Shared Key)

This section describes the process of configuring a site-to-site connection using a **shared key** style OpenVPN tunnel.

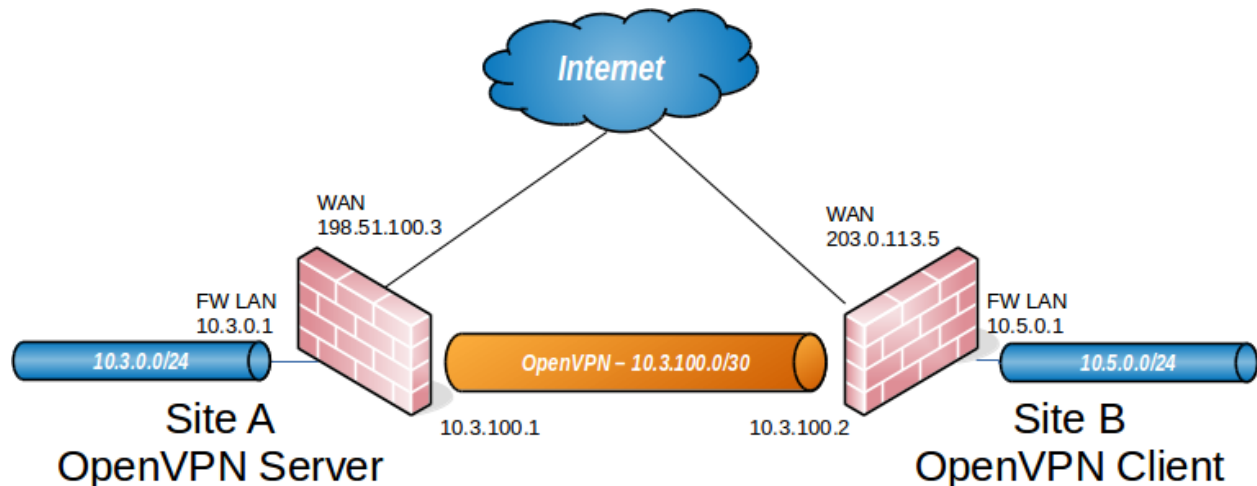


Fig. 20.10: OpenVPN Example Site-to-Site Network

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.

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*.

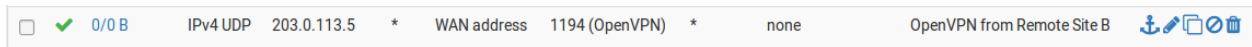


Fig. 20.11: OpenVPN Example Site-to-Site WAN Firewall Rule

- 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.

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.

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](#).

Site-to-Site Example Configuration (SSL/TLS)

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

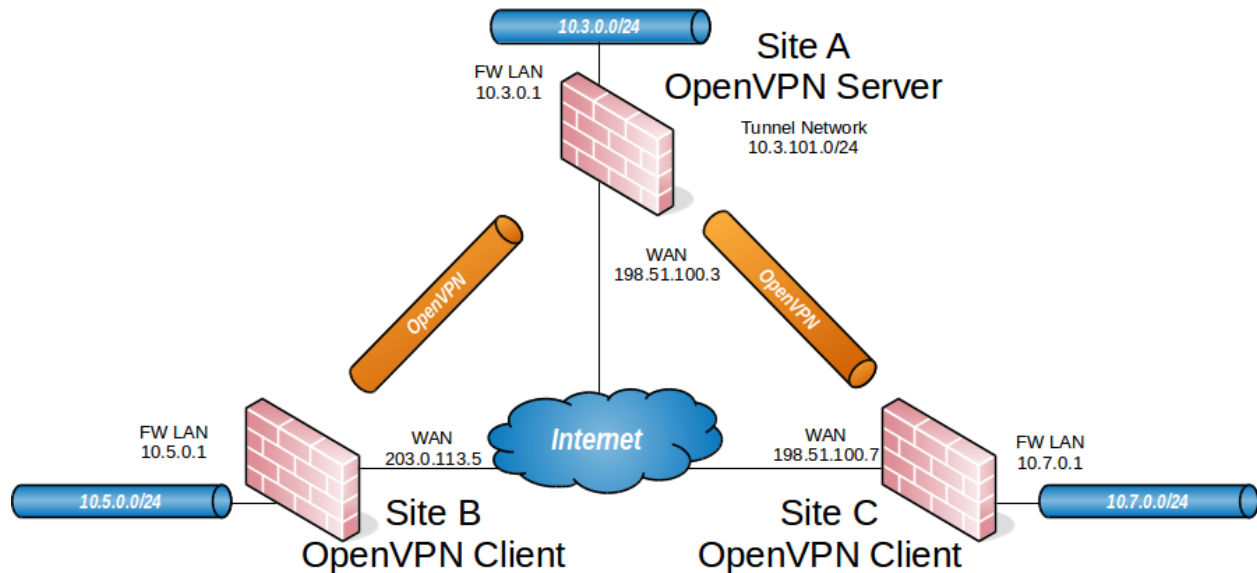


Fig. 20.12: OpenVPN Example Site-to-Site SSL/TLS Network

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.

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 (`.cert`) 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.

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.

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](#).

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.

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.

Remote Access UDP:1194 Client Connections					
Common Name	Real Address	Virtual Address	Connected Since	Bytes Sent	Bytes Received
jump	198.51.100.6:3037	10.3.201.2	Wed Jun 8 15:09:55 2016	19 KiB	16 KiB
▶ Running					
Show Routing Table - Display OpenVPN's internal routing table for this server.					

Fig. 20.13: OpenVPN Status for SSL/TLS Server With One Connected Client

For OpenVPN clients, the status indicates whether a connection is pending or active.

Peer to Peer Server Instance Statistics							
Name	Status	Connected Since	Virtual Address	Remote Host	Bytes Sent	Bytes Received	Service
P2P-SK-Server UDP:1194	up	Thu May 26 13:31:11 2016	172.16.9.1	198.51.100.111	7.28 MiB	7.28 MiB	▶
P2P-SSL-Server#2 UDP:1196					0 B	0 B	▶

Client Instance Statistics							
Name	Status	Connected Since	Virtual Address	Remote Host	Bytes Sent	Bytes Received	Service
P2P-SK-Client UDP	reconnecting; ping-restart	Wed Jun 8 15:30:43 2016			0 B	0 B	▶

Fig. 20.14: OpenVPN Status Showing a Server that is up, one waiting for a Connection, and a Client Attempting to Reconnect

Permitting traffic to the OpenVPN server

After setting up an OpenVPN server, a firewall rule to permit traffic to the OpenVPN server is required.

- 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
- 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*.

☐		0/54 KiB	IPv4 UDP	*	*	WAN address	1194 (OpenVPN)	*	none	Allow traffic to OpenVPN server			

Fig. 20.15: 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 system from the remote side, not traffic leaving from the server side, so craft the rules accordingly. In cases when pfSense is used 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.

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.

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 > (assign)**
- 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.

Interface Assignments		Interface Groups	Wireless	VLANs	QinQs	PPPs	GREs	GIFs	Bridges	LAGGs
Interface	Network port									
WAN	igb1 (00:08:a2:09:95:b6)									
LAN	igb0 (00:08:a2:09:95:b5)									
OPT1	ovpns1 (Site-to-Site VPN)									
Available network ports:		igb2 (00:08:a2:09:95:b1)								 Add

Fig. 20.16: 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)

NAT with OpenVPN Connections

For many advanced NAT Scenarios using OpenVPN, assigning the interface is required as covered in [Assigning OpenVPN Interfaces](#)

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.

For example, if 10.3.0.0/24 is the LAN on both sides of a VPN then hosts on a 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 *Interface assignment and configuration*, 1:1 NAT can be applied.

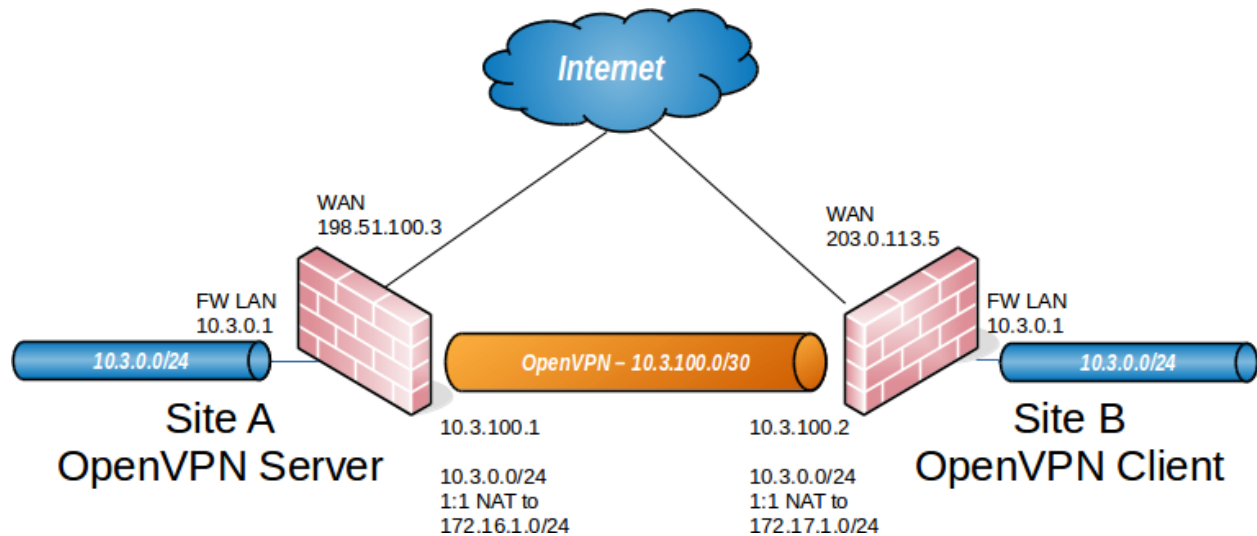


Fig. 20.17: Site to Site with Conflicting Subnets

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.

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.

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

Edit NAT 1:1 Entry			
Disabled	☐ Disable this rule When disabled, the rule will not have any effect.		
No BINAT (NOT)	☐ Do not perform binat for the specified address Excludes the address from a later, more general, rule.		
Interface	OPT1 Choose which interface this rule applies to. In most cases "WAN" is specified.		
External subnet IP	172.16.1.0 Enter the external (usually on a WAN) subnet's starting address for the 1:1 mapping. The subnet mask from the internal address below will be applied to this IP address.		
Internal IP	☐ Not Invert the sense of the match.	Network Type	10.3.0.0 / 24 Address/mask
Enter the internal (LAN) subnet for the 1:1 mapping. The subnet size specified for the internal subnet will be applied to the external subnet.			
Destination	☐ Not Invert the sense of the match.	Any Type	/ Address/mask
The 1:1 mapping will only be used for connections to or from the specified destination. Hint: this is usually "Any".			
Description	1:1 NAT for OpenVPN A description may be entered here for administrative reference (not parsed).		
NAT reflection	Use system default		

Fig. 20.18: Site A 1:1 NAT Configuration

Edit NAT 1:1 Entry			
Disabled	☐ Disable this rule When disabled, the rule will not have any effect.		
No BINAT (NOT)	☐ Do not perform binat for the specified address Excludes the address from a later, more general, rule.		
Interface	OPT1 Choose which interface this rule applies to. In most cases "WAN" is specified.		
External subnet IP	172.17.1.0 Enter the external (usually on a WAN) subnet's starting address for the 1:1 mapping. The subnet mask from the internal address below will be applied to this IP address.		
Internal IP	☐ Not Invert the sense of the match.	Network Type	10.3.0.0 / 24 Address/mask
Enter the internal (LAN) subnet for the 1:1 mapping. The subnet size specified for the internal subnet will be applied to the external subnet.			
Destination	☐ Not Invert the sense of the match.	Any Type	/ Address/mask
The 1:1 mapping will only be used for connections to or from the specified destination. Hint: this is usually "Any".			
Description	1:1 NAT for OpenVPN A description may be entered here for administrative reference (not parsed).		
NAT reflection	Use system default		

Fig. 20.19: Site B 1:1 NAT Configuration

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, 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

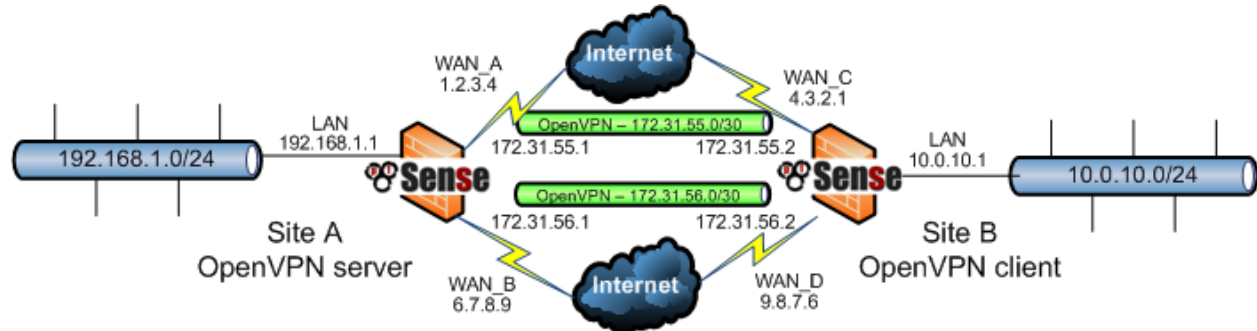


Fig. 20.20: Example OpenVPN Setup Involving OSPF Across Multiple WANs

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 Quagga_OSPF package from **System > Packages, Available Packages** tab on **both** firewalls.
- Navigate to **Services > Quagga OSPFd, 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**

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.

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.

Bridged OpenVPN Connections

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.

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.

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 > (assign), 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](#).

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.

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”.)

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.

Now if an OpenVPN client is pointed to the public address, it will connect and work fine, 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 may result in reduced VPN performance.

Controlling Client Parameters via RADIUS

When using RADIUS as an authentication source for a VPN, pfSense supports receiving some client configuration parameters from the RADIUS server as reply attributes. The following values may be specified:

Cisco-AVPair inacl= Inbound firewall rules to govern traffic from the client to the server. Given in Cisco-style ACL format (e.g. `permit tcp from any to any`) subnet masks are specified wildcard style.

Cisco-AVPair outacl= 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.

Troubleshooting OpenVPN

If problems are encountered when trying to use OpenVPN, consult this section for information on troubleshooting common issues.

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](#).

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.

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 [Using tcpdump from 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.

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.

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.

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.

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.

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](#).

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.

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 0xa
      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.

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 [Site-to-Site Example Configuration \(SSL/TLS\)](#), especially the client-specific overrides and `iroutes`.

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.

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.

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.

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 it 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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the September 2014 Hangout on Advanced OpenVPN Concepts and the September 2015 Hangout on Remote Access VPNs

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](#).

L2TP VPN

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.

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.

L2TP Server Configuration

To use L2TP, first browse to **VPN > L2TP**. Select **Enable L2TP server**.

Interface

The **Interface** setting controls where the L2TP daemon will bind and listen for connections. This is typically the WAN interface accepting inbound connections.

IP Addressing

Before starting, determine which IP addresses to use for the L2TP server and clients and how 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 *16* has been selected.

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.

Enable L2TP	
Enable	☒ Enable L2TP server
Configuration	
Interface	WAN
Server address	10.3.177.1 Enter the IP address the L2TP server should give to clients for use as their "gateway". Typically this is set to an unused IP just outside of the client range. NOTE: This should NOT be set to any IP address currently in use on this firewall.
Remote address range	10.3.177.128 / 25 Specify the starting address for the client IP address subnet.
Number of L2TP users	13

Fig. 21.1: L2TP IP Addressing

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 [RADIUS Authentication with Windows Server](#)) 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. A second RADIUS server can also be added in case the first one fails. 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.

Save changes to start L2TP server

After filling in the aforementioned items, click **Save**. This will save the configuration and launch the L2TP server.

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.

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.

FloatingWANLANDMZWAN2L2TP VPNIPsecOpenVPN

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	☒	0/0 B	IPv4 *	*	*	*	*	none			   

Fig. 21.2: L2TP VPN Firewall Rule

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.

Configuration

Users

L2TP Users

Username	IP address	Actions

Fig. 21.3: 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.

User		
Username	someguy	
Password	••••••	
		••••••
		Confirm
IP Address		
To assign the user a specific IP address, enter it here.		

Fig. 21.4: Adding a L2TP User

- 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 .

L2TP with IPsec

On current versions of pfSense, L2TP/IPsec may be configured for mobile clients, though it is not a configuration we recommend.

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:

[Example IKEv2 Server Configuration](#) 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.

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 on pfSense
- 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*
- 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 `aaabbbbccc` – ideally one a lot longer, more random, and secure!
- Click **Save**
- Click **Apply Changes**

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**

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**

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*)

L2TP Troubleshooting

This section covers troubleshooting steps for the most common problems users encounter with L2TP.

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.

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.

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.

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*

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.

pfSense 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:

[Example IKEv2 Server Configuration](#) 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](#).

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.

TRAFFIC SHAPER

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.

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.

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.

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.

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.

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.

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), ovps(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.

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.

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.

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.

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](#).

Class-Based Queuing (CBQ)

Class-Based Queuing, or CBQ, is similar to HFSC in that it 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.

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)

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.

Configuring the ALTQ Traffic Shaper With the Wizard

We recommend configuring the traffic shaper 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.

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.

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. 22.1: Entering the Interface Count

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

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.

Setup connection speed and scheduler information for interface LAN #1	
Interface & Scheduler	LAN
Interface & Scheduler	PRIQ
Setup connection speed and scheduler information for interface WAN#1	
Interface & Scheduler	WAN
Interface & Scheduler	PRIQ
Upload	1
Upload	Mbit/s
Download	10
Download	Mbit/s

Fig. 22.2: Shaper Configuration

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

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

Voice over IP	
Voice over IP	
enable	☒ Prioritize Voice over IP traffic.
VOIP specific settings	
Provider	Generic (lowdelay) Choose Generic if the provider isn't listed.
Upstream SIP Server	203.0.113.49 (Optional) If this is chosen, the provider field will be overridden. This allows providing the IP address of the remote PBX or SIP Trunk to prioritize. NOTE: A Firewall Alias can also be used in this location.
Connection WAN #1	
Upload	
Units	Mbit/s
Connection LAN #1	
Download	
Units	Mbit/s

Fig. 22.3: Voice over IP

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

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.

Penalty Box	
Penalty Box	
Enable	☒ Penalize IP or Alias This will lower the priority of traffic from this IP or alias.
PenaltyBox specific settings	
Address	192.168.1.15 This allows just providing the IP address of the computer(s) to penalize. NOTE: A Firewall Alias can also be used in this location.
Bandwidth	10
Bandwidth	% The desired limit to apply.

Fig. 22.4: Penalty Box

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

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.

Peer to Peer networking

Peer to Peer networking

Enable

☐ Lower priority of Peer-to-Peer traffic

This will lower the priority of P2P traffic below all other traffic. Please check the items to prioritize lower than normal traffic.

p2p Catch all

p2pCatchAll

☐ When enabled, all uncategorized traffic is fed to the p2p queue.

Bandwidth

Bandwidth

%

The desired limit to apply.

Enable/Disable specific P2P protocols

Aimster

☐ Aimster and other P2P using the Aimster protocol and ports

BitTorrent

☐ Bittorrent and other P2P using the Torrent protocol and ports

BuddyShare

☐ BuddyShare and other P2P using the BuddyShare protocol and ports

Fig. 22.5: Peer-to-Peer Networking

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

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.

Network Games	
Network Games	
Enable	☒ Prioritize network gaming traffic This will raise the priority of gaming traffic to higher than most traffic.
Enable/Disable specific game consoles and services	
BattleNET	☐ Battle.net - Virtually every game from Blizzard publishing should match this. This includes the following game series: Starcraft, Diablo, Warcraft. Guild Wars also uses this port.
EAOrigin	☐ EA Origin Client - Some PC games by EA use this.
GameForWindowsLive	☐ Games for Windows Live
PlayStationConsoles	☐ PlayStation Consoles - This should cover all ports required for the Playstation 4, Playstation, PS Vita
Steam	☒ Steam Game Client (Includes: America's Army 3, Counter-Strike: Source, Counter-Strike: Global Offensive, Half-Life 2, COD: Black Ops Series, Borderlands 2, Natural Selection 2, Left 4 Dead Series, Portal 2 and many other games on the Steam)
WiiConsoles	☒ Wii Consoles - Wii, Wii U, DS and 3DS
XboxConsoles	☐ Xbox Consoles - Xbox 360 and Xbox One
Enable/Disable specific games	
ARMA2	☐ ARMA 2

Fig. 22.6: Network Games

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

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**

Raise or lower other Applications

Raise or lower other Applications

Enable

☒ Other networking protocols

This will help raise or lower the priority of other protocols higher than most traffic.

Remote Service / Terminal emulation

AppleRemoteDesktop

Default priority

MSRDP

Higher priority

PCAnywhere

Default priority

VNC

Higher priority

Messengers

AIM

Default priority

Fig. 22.7: Raise or Lower Other Applications

Shaper Wizard and IPv6

The shaper wizard creates rules for IPv4 traffic only. Rules can be manually adjusted or cloned and set for IPv6.

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.

Status Queues								
Queue	Statistics		PPS	Bandwidth	Borrows	Suspends	Drops	Length
	PPS							
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. 22.8: Basic WAN Queues

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.

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.

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.

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.

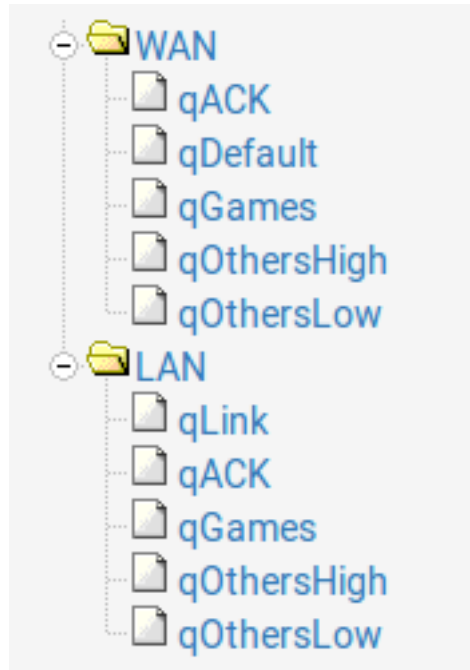


Fig. 22.9: Traffic Shaper Queues List

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:

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.

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.

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.

Floating

WAN

LAN

WANv6

OpenVPN

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐		0 / 420 B	IPv4	*	*	*	27000 - 27030	*	qGames	m_Game Steam-game-udp outbound	
☐		0 / 420 B	IPv4	*	*	*	27000 - 27030	*	qACK/qGames	m_Game Steam-game-tcp outbound	
☐		0 / 420 B	IPv4	*	*	*	27015 - 27030	*	qGames	m_Game Steam-hltv outbound	
☐		0 / 420 B	IPv4	*	*	*	4380	*	qGames	m_Game Steam-1 outbound	
☐		0 / 420 B	IPv4	*	*	*	1200	*	qGames	m_Game Steam-2 outbound	
☐		0 / 420 B	IPv4	*	*	*	3478 - 3480	*	qGames	m_Game Steam-voice outbound	
☐		0 / 420 B	IPv4	*	*	*	6667	*	qACK/qGames	m_Game Wii-Consoles-TCP-1 outbound	

Fig. 22.10: Traffic Shaper Rules List

- 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.

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

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.

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.

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.

Limiters and IPv6

Limiters work with IPv6, though it requires separate IPv4 and IPv6 rules to apply limiters properly.

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](#).

Warning: At this time, Limiters are not compatible with `pfsync` state synchronization. For firewall clusters using High Availability, state synchronization must be disabled when using limiters.

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.

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 /28 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”.

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.

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.

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.

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.

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.

Troubleshooting Shaper Issues

Traffic Shaping/QoS is a tricky topic, and can prove difficult to get right the first time. This section covers several common pitfalls.

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.

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.

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 pfSense forums](#). 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.

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.

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.

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”.

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.

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.

SERVER LOAD BALANCING

Server Load Balancing Configuration Options

Pools

To configure Pools:

- Navigate to **Services > Load Balancer**
- Click the **Pools** tab
- Click  **Add** to add a new pool
- Configure the pool options as explained below:

Name A name for the Pool. The name is how the pool is referenced when configuring the Virtual Server that will use this pool. This name must adhere to the same limits as an alias or interface name. Letters and numbers only, the only allowed separator is an underscore. .. note:: This name cannot be the same as an existing alias.

Mode Select *Load Balance* to balance load between all servers in the pool, or *Manual Failover* to always use the servers in the Enabled list, and they can be manually moved between an enabled and disabled state.

Description A optional longer description for the Pool.

Port This is the port the servers are listening on internally. This can be different from the external port, which is defined later in the Virtual Server configuration. An alias may also be used to define multiple ports, however, if an alias is used it must use the same port alias here and in the Virtual Server configuration.

Retry This defines the number of times a server will be contacted by the monitor before being declared down.

Monitor This defines the type of monitor to use, which is how the load balancer determines if the servers are up and usable. Selecting *TCP* will make the balancer connect to the port previously defined in **Port**, and if it cannot connect to that port, the server is considered down. Choosing *ICMP* will instead monitor the defined servers by sending an ICMP ping, and will mark them down if they do not respond. There are many more types of monitors, and they can be customized. They are covered in more detail later in the chapter.

Server IP Address This is where the internal IP addresses of the servers in the pool are listed. Enter them one at a time, clicking **Add to pool** afterwards.

Current Pool Members This field shows the list of servers in this pool. A server can be removed from the pool by clicking on its IP address and then clicking **Remove**. There are two lists in

this section, **Pool Disabled**, and **Enabled (default)**. The servers in the **Enabled (default)** list are active and used, servers in the **Pool Disabled** list are never used. The **Pool Disabled** list is primarily used with *Manual Failover* mode. Servers can be moved between the lists by selecting

them and clicking  or .

- Click **Save**

If automatic failover is required, create a second pool to be used as a **Fall Back Pool**, containing the backup set of server IP addresses.

Virtual Servers

To configure a Virtual Server to handle client connections:

- Navigate to **Services > Load Balancer**
- Click the **Virtual Servers** tab
- Click  **Add** to add a new Virtual Server
- Configure the Virtual Server options as explained below:

Name A name for the Virtual Server. This is for reference, but must also adhere to the same limits as an alias or interface name. Letters and numbers only, the only allowed separator is an underscore. No spaces or slashes.

Description An optional longer description for the Virtual Server. This is for reference purposes only, and does not have any formatting limits.

IP Address This is where IP addresses are entered for use by the Virtual Server. This is usually the WAN IP address or a Virtual IP address on WAN. It must be a static IP address. A CARP VIP may also be used for a high availability setup. For more information on high availability and CARP VIPs, refer to [High Availability](#). An IP Alias VIP may be used, or a Proxy ARP VIP (TCP mode only). Furthermore, an Alias may also be used here to specify multiple IP addresses upon which this Virtual Server may accept connections.

Note: In TCP mode, the IP addresses specified here are **not** bound at the OS level, meaning that `relayd` as a daemon is not bound and listening on these ports directly.

Port This is the port upon which the Virtual Server will accept connections. It can be different from the port used by the pool servers internally. An alias can be used to define multiple ports, however, if the same port alias must be used here and in the Pool configuration.

Virtual Server Pool This is where the previously configured pool is selected. The connections to the **IP Address** and **Port** defined on this screen will be directed to the IP addresses and ports configured in the pool.

Fall Back Pool This is the alternate pool that clients are directed to if all the servers in the primary pool are down. If there is no alternate server, leave this set to *None*, though the result will be inaccessibility if all the servers in the pool are down. If nothing else, to avoid having the server be down entirely, setup a simple web server to return a basic maintenance page for any request and use it as the fall back pool.

Relay Protocol The Relay Protocol can be either *TCP* or *DNS*, depending on what this relay will be doing.

- In *TCP* mode, `relayd` acts like an enhanced port forward, directing connections as though they were hitting a traditional NAT rule. Servers will see the original source IP address of the client, it does not act as a proxy.
- In *DNS* mode, `relayd` acts as a DNS proxy. It will balance the load over multiple DNS servers, but the original client IP address is lost. Pool servers will see the firewall as the source of the DNS query. Keep this in mind when setting up views or source-based query restrictions on DNS servers involved in load balancing.

- Click **Submit**
- Click **Apply Changes**

Warning: If all Virtual Server Pool members and Fall Back Pool members are down, `relayd` will act as though the Load Balancer is not handling connections for the Virtual Server IP address and port. If the IP address and port used are also used by another service or NAT rule, it could be accidentally exposed to clients.

Monitors

There are five basic pre-defined Monitor types: ICMP, TCP, HTTP, HTTPS, and SMTP. Additional custom types may be added to better detect specific types of failures.

Pre-defined Monitors

The pre-defined monitors are included in the default configuration and are:

ICMP Sends an ICMP echo request to the target server and expects an ICMP echo reply.

TCP Attempts to open a TCP port connection to the target IP address and port. If the port can be opened (3-way TCP handshake) then it succeeds, if it connection is refused or timed out, it fails.

HTTP & HTTPS Attempts to open a connection to the server and request the URL / using HTTP or HTTPS, whichever is selected. If a 200 response code is returned, it is OK. Otherwise, it is considered a failure.

SMTP Opens a connection to the defined port and sends the string `EHLO nosuchhost`. If the server replies with any message starting with `250-`, it is considered OK. Other responses are considered a failure.

Creating Custom Monitors

The included monitors are not sufficient for the needs of a site, or they need tweaking, then custom monitors may be created. Most monitor types have their own specific settings that can be customized as needed.

To create a new monitor: * Navigate to **Services > Load Balancer** * Click the **Monitors** tab * Click  **Add** to add a new Monitor * Configure the Monitor options as explained below:

Name A name for the Monitor. This is for reference, but must also adhere to the same limits as an alias or interface name. Letters and numbers only, the only allowed separator is an underscore. No spaces or slashes.

Description An optional longer description for the Monitor. This is for reference purposes only, and does not have any formatting limits.

The remaining options vary based on the selected Type.

ICMP & TCP No extra options. Any custom monitor using these types will behave identically to the pre-defined monitor of the same name.

HTTP & HTTPS These behave identically to each other, the only difference is whether or not encryption is used to talk to the target server. These each have three options to control the behavior of the monitor:

Path The Path defines the path section of the URL sent to the server. If the site contains mostly dynamic content, or the base URL does a redirect, it is best to set this to a full path to a static piece of content, such as an image, that is unlikely to move or change.

Host If the server runs multiple virtual hosts, this field defines which hostname is sent with the request so that the expected response can be received.

HTTP Code This defines the response expected from the server, given the request to the Host/Path. Most commonly this would be set to *200 OK*, but if the server uses another return code that would be expected as a healthy response to this query, choose it here. If the return code is unknown, inspect the server logs to find what codes are returned to the client for each request.

Send/Expect This type of monitor opens a connection to the defined port and sends a string and expects the specified response. The most common example is the SMTP monitor discussed previously. The options are:

Send String The string sent to the server after a connection is made to its port.

Expect String If the response from the server does not start with this string, then it is considered down.

- Click **Save**

Settings

In addition to the per-pool or per-server options, there are also a few global options that control the behavior of relayd. These settings are under **Services > Load Balancer** on the **Settings** tab:

Timeout The global timeout in milliseconds for checks. Leave blank to use the default value of 1000 ms (1 second). If a loaded server pool takes longer to respond to requests, increase this timeout.

Interval The interval in seconds at which the member of a pool will be checked. Leave blank to use the default interval of 10 seconds. To check the servers more (or less) frequently, adjust the timing accordingly.

Prefork Number of processes used by `relayd` for handling inbound connections to relays. This option is only active for relays using *DNS mode*. It does not have any effect on TCP mode since that uses a redirect, not a relay. Leave blank to use the default value of 5 processes. If the server is busy, increase this amount to accommodate the load.

Firewall rules

The last step in configuring Load Balancing is to configure firewall rules to allow traffic to the pool.

For *TCP* mode, the firewall rules must permit traffic to the internal private IP addresses of the servers, the same as with NAT rules, as well as the port they are listening on internally. Create an alias for the servers in the pool to make the process easier, and create a single firewall rule on the interface where the traffic destined to the pool will be initiated (usually WAN) allowing the appropriate source (usually *any*) to a destination of the alias created for the pool.

A specific example of this is provided in [Configuring firewall rules](#). For more information on firewall rules, refer to [Firewall](#).

For *DNS* mode, firewall rules must allow traffic directly to the Virtual Server IP address and port, not the pool servers.

Sticky connections

There is one additional configuration option available for server load balancing, under **System > Advanced**, on the **Miscellaneous** tab. Under **Load Balancing**, called **Use sticky connections**. Checking this box will attempt to send clients with an active connection to the pool server to the same server for any subsequent connections.

Once the client closes all active connections, and the closed state times out, the sticky connection is lost. This may be desirable for some web load balancing configurations where client requests must only go to a single server, for session or other reasons. This isn't perfect, as if the client's web browser closes all TCP connections to the server after loading a page and sits there for 10 minutes or more before loading the next page, the next page may be served from a different server. Generally this isn't an issue as most web browsers won't immediately close a connection, and the state exists long enough to not make it a problem, but if the site is strictly reliant on a specific client never getting a different server in the pool regardless of how long the browser sits there inactive, look for a different load balancing solution. There is a box under the option to control the **Source Tracking Timeout** which can allow the knowledge of the client/server relationship to persist longer.

Warning: Sticky is generally unreliable for this purpose and can also have other unintended side effects. Full-featured proxy packages such as HAProxy have much better mechanisms and options for maintaining client/server relationships.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the January 2015 Hangout on Server Load Balancing and Failover, which includes information on configuring HAProxy.

There are four areas of configuration for the server load balancer:

1. **Pools** define collections of servers to be used, which port they use, and the monitoring method.
2. **Virtual Servers** define the IP address and port for accepting user connections, and the appropriate pool to direct the incoming traffic destined to that IP address and port.
3. **Monitors** are used to create custom monitoring methods to determine if pool servers are working and usable.
4. The **Settings** tab contains global options that alter how the load balancer operates.

In a typical example, there is a **Virtual Server** to accept user connections, and it contains several servers in a **Pool**. The **Pool** utilizes a **Monitor** for each server to determine if it is capable of accepting user connections.

A Virtual Server can have a regular and a **Fall Back Pool** to use if all members of the regular **Virtual Server Pool** are down. This can be leveraged to present a maintenance or outage page, for example.

Web Server Load Balancing Example Configuration

This section shows how to configure the Load Balancer from start to finish for load balanced environment with two web servers.

Example network environment

Figure [Server Load Balancing Example Network](#) shows the example environment configured in this section. It consists of a single firewall, using its WAN IP address for the pool, with two web servers on a DMZ segment.

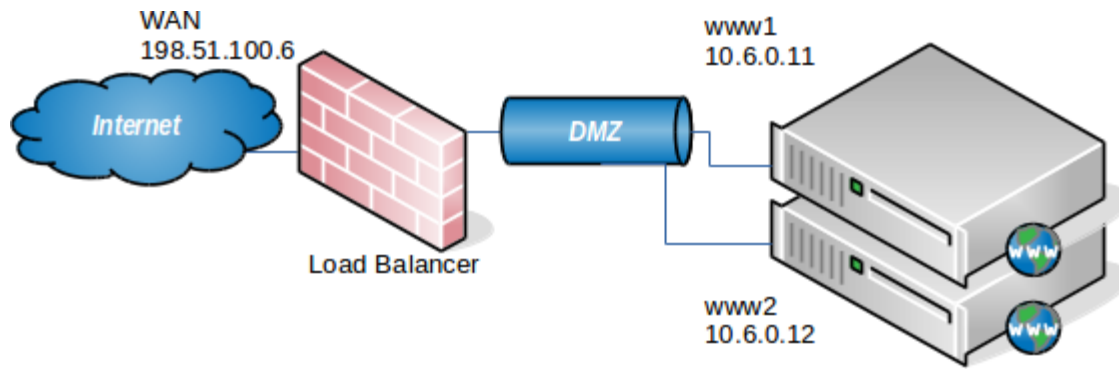


Fig. 23.1: Server Load Balancing Example Network

Configuring pool

To configure the pool:

- Navigate to **Services > Load Balancer**
- Click the **Pools** tab
- Click  **Add** to create a new pool
- Configure the pool as shown in Figure *Pool Configuration*, which uses the following settings:
 - Name** WebServers
 - Mode** Load Balance
 - Description** Web server Pool
 - Port** 80
 - Retry** 5
 - Pool Members** Add both web servers (10.6.0.11 and 10.6.0.12) using an *HTTP Monitor*
- Click **Save**

Configuring virtual server

- Click the **Virtual Servers** tab
- Click  **Add** to add a new virtual server
- Configure the Virtual Server as shown in Figure *Virtual Server Configuration*, which uses the following settings:
 - Name** WebVirtualServer
 - Description** Web Server
 - IP Address** The firewall's WAN IP address, 198.51.100.6
 - Port** 80
 - Virtual Server Pool** WebServers
 - Fall Back Pool** None
- Click **Submit**

Add/Edit Load Balancer - Pool Entry

Name

WebServers

Mode

Load Balance

Description

Port

80

This is the port the servers are listening on. A port alias listed in Firewall -> Aliases may also be specified here.

Retry

5

Optionally specify how many times to retry checking a server before declaring it down.

Add Item to the Pool

Monitor

HTTP

Server IP Address

IP Address

+ Add to pool

Current Pool Members

Members

Disabled

Remove

>> Move to enabled list

10.6.0.11
10.6.0.12

Enabled (Default)

Remove

<< Move to disabled list

Fig. 23.2: Pool Configuration

Edit Load Balancer - Virtual Server Entry

Name

WebVirtualServer

Description

Web Server

IP Address

198.51.100.6

This is normally the WAN IP address for the server to listen on. All connections to this IP and port will be forwarded to the pool cluster. A host alias listed in Firewall -> Aliases may also be specified here.

Port

80

Port that the clients will connect to. All connections to this port will be forwarded to the pool cluster. If left blank listening ports from the pool will be used. A port alias listed in Firewall -> Aliases may also be specified here.

Virtual Server Pool

WebServers

Fall-back Pool

None

Relay Protocol

TCP

Fig. 23.3: Virtual Server Configuration

23.2. Web Server Load Balancing Example Configuration

479

- Click **Apply Changes**

Warning: In this example, if both of the pool servers are down, the Virtual Server is inaccessible. The firewall will act as if no Virtual Server is configured. If something on the firewall is bound to port 80, clients will reach that instead. This includes the built-in Web GUI redirect for port 80, so that should be disabled under **System > Advanced** on the **Admin Access** tab.

Configuring firewall rules

Firewall rules must be configured to allow access to the servers in the pool. The rules must allow the traffic to the internal IP addresses and port being used, and no rules are necessary for the outside IP Address and Port used in the virtual server configuration.

Create an alias containing all the servers in the pool, so access can be allowed with a single firewall rule.

- Navigate to **Firewall > Aliases**

- Click  **Add** to add an alias.

- Use the following settings:

Name `www_servers`

Type *Hosts*

Hosts The IP addresses of both web servers: `10.6.0.11` and `10.6.0.12`

- Click **Save**
- Click **Apply Changes**

Figure *Alias for Web Servers* shows the alias used for this example configuration, containing the two web servers.

Properties			
Name	www_servers The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".		
Description	Web Servers A description may be entered here for administrative reference (not parsed).		
Type	Host(s)		
Host(s)			
Hint	Enter as many hosts as desired. Hosts must be specified by their IP address or fully qualified domain name (FQDN). FQDN hostnames are periodically re-resolved and updated. If multiple IPs are returned by a DNS query, all are used. An IP range such as 192.168.1.1-192.168.1.10 or a small subnet such as 192.168.1.16/28 may also be entered and a list of individual IP addresses will be generated.		
IP or FQDN	10.6.0.11 / 32	www1	Delete
	10.6.0.12 / 32	www2	Delete

Fig. 23.4: Alias for Web Servers

Next, create a firewall rule using that alias: * Navigate to **Firewall > Rules** * Change to the tab for the interface where connections will enter (e.g. **WAN**) * Click  **Add** to start a new rule at the top of the list * Use the following

settings:

Interface WAN
Protocol TCP
Source any
Destination Type Single Host or Alias
Destination Address www_servers
Destination Port Range HTTP
Description Allow to Web Server

- Click **Save**
- Click **Apply Changes**

Figure *Adding Firewall Rule for Web Servers* shows a snippet of the firewall rule added for this configuration. The options not shown are left at their defaults.

Edit Firewall Rule

Action

Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled

☐ Disable this rule

Set this option to disable this rule without removing it from the list.

Interface

WAN

Choose the interface from which packets must come to match this rule.

Address Family

IPv4

Select the Internet Protocol version this rule applies to

Protocol

TCP

Choose which IP protocol this rule should match.

Source

Source

☐ Invert match.

any

Source Address

/

Display Advanced

☒ Display Advanced

Destination

Destination

☐ Invert match.

Single host or alias

www_servers

/

Destination port range

From

HTTP (80)

Custom

To

HTTP (80)

Custom

Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

Fig. 23.5: Adding Firewall Rule for Web Servers

Figure *Firewall Rule for Web Servers* shows the rule as it appears in the list.

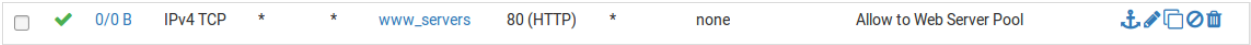


Fig. 23.6: Firewall Rule for Web Servers

Viewing load balancer status

Now that the load balancer is configured, to view its status, browse to **Status > Load Balancer** and click the **Virtual Servers** tab. This page displays the status of the server as a whole, typically listed as either **Active** or **Down**.

The **Pools** tab shows an individual status for each member of a Pool (as shown in Figure *Pool Status*). The row for a server is green if it is online, and red if the server is offline.

Additionally, each server in the pool has a checkbox next to it. Servers that are checked are active in the pool, and unchecked servers are disabled in the pool, the same as moving them between the enabled and disabled list on the pool editing page. To disable a server: Uncheck it, then click **Save**.

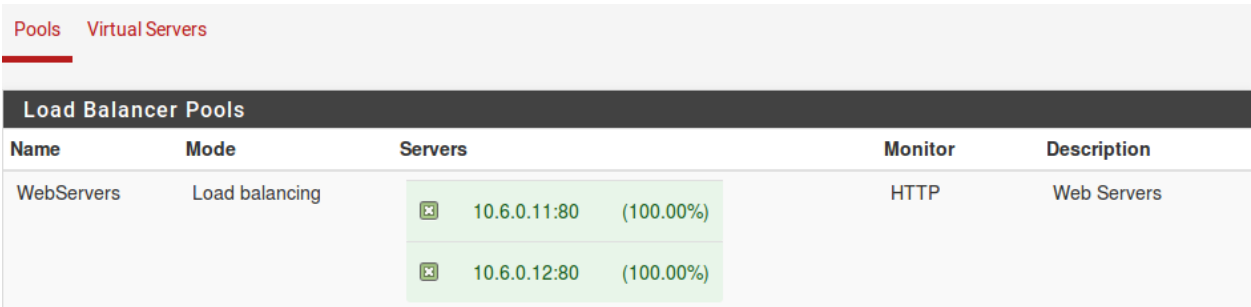


Fig. 23.7: Pool Status

If the web server service is stopped on one of the servers, or if the server is removed from the network entirely if using ICMP monitors, the status updates to Offline and the server is removed from the pool.

Verifying load balancing

To verify load balancing, `curl` is the best option to ensure the web browser cache and persistent connections do not affect the results of testing. `curl` is available for every OS imaginable and can be downloaded from the [curl website](#). To use it, simply run:

```
curl http://mysite
```

In that command, replace `198.51.100.6` with either the IP address or hostname of the site. This **must** be tested from outside the network (e.g. from a remote network or client on WAN). The following illustrates an example of testing with `curl` from the WAN side:

```
# curl http://198.51.100.6
This is server www2 - 10.6.0.12
# curl http://198.51.100.6
This is server www1 - 10.6.0.11
```

When initially testing load balancing, configure each server to return a page specifying its hostname, IP address, or both, so it is made obvious which server is responding to the request. If sticky connections is not enabled, a different server will respond to each request.

Troubleshooting Server Load Balancing

This section describes how to identify, troubleshoot, and resolve the most common issues encountered by users with server load balancing.

Connections not being balanced

Connections not being balanced is most always a failure of the testing methodology being used, and is usually specific to HTTP. Web browsers will commonly keep connections to a web server open, and hitting refresh re-uses the existing connection. A single connection will never be changed to another balanced server. Another common issue is the web browser cache, where the browser never actually requests the page again. It is preferable to use a command line tool such as `curl` for testing of this nature, because it ensures the test is not impacted by the problems inherent in testing with web browsers. `curl` has no cache, and opens a new connection to the server each time it is run. More information on `curl` can be found in [Verifying load balancing](#).

If sticky connections are enabled, ensure testing is performed from multiple source IP addresses. Tests from a single source IP address will go to a single server unless a long period of time elapses between connection attempts.

Down server not marked as offline

If a server goes down but is not marked as offline, it is because the monitoring performed by the load balancing daemon believes it is still up and running. If using a TCP monitor, the TCP port must still be accepting connections. The service on that port could be broken in numerous ways and still answer TCP connections. For ICMP monitors, this problem is exacerbated, as servers can be hung or crashed with no listening services at all and still answer to pings.

Live server not marked as online

If a server is online, but not marked as online, it is because it isn't online from the perspective of the load balancing daemon monitors. The server must answer on the TCP port used or respond to pings sourced from the IP address of the firewall interface closest to the server.

For example, if the server is on the LAN, the server must answer requests initiated from the LAN IP address of the firewall. To verify this for ICMP monitors, browse to **Diagnostics > Ping** and ping the server IP address using the interface where the server is located.

For TCP monitors, use **Diagnostics > Test Port**, and choose the firewall's LAN interface as the source, and the web server IP address and port as the target.

Another way to test is from a shell prompt on the firewall, either using the console or ssh menu option 8 and the `nc` command:

```
# nc -vz 10.6.0.12 80
nc: connect to 10.6.0.12 port 80 (tcp) failed: Operation timed out
```

And here is an example of a successful connection:

```
# nc -vz 10.6.0.12 80
Connection to 10.6.0.12 80 port [tcp/http] succeeded!
```

If the connection fails, troubleshoot further on the web server.

Unable to reach a virtual server from a client in the same subnet as the pool server

Client systems in the same subnet as the pool servers will fail to properly connect using this load balancing method. `relayd` forwards the connection to the web server with the source address of the client intact. The server will then try to respond directly to the client. If the server has a direct path to the client, e.g. through a locally connected NIC in the same subnet, it will not flow back through the firewall properly and the client will receive the reply from the server's local IP address and not the IP address in `relayd`. Then, due to the fact that the server IP address is incorrect from the perspective of the client, the connection is dropped as being invalid.

One way around this is by using manual outbound NAT and crafting a manual outbound NAT rule so that traffic leaving the internal interface (LAN) coming from the LAN subnet, going to the web servers, gets translated to the interface address of LAN. That way the traffic appears to originate from the firewall, and the server will respond back to the firewall, which then relays the traffic back to the client using the expected addresses. The original client source IP address is lost in the process, but the only other viable solution is to move the servers to a different network segment.

Two types of load balancing functionality are available in pfSense: Gateway and Server. Gateway load balancing enables distribution of Internet-bound traffic over multiple WAN connections. For more information on this type of load balancing, see [Multiple WAN Connections](#). Server load balancing manages incoming traffic so it utilizes multiple internal servers for load distribution and redundancy, and is the subject of this chapter.

Server load balancing allows traffic to be distributed between multiple internal servers. It is most commonly used with web servers and SMTP servers though it can be used for any TCP service or for DNS.

While pfSense has replaced high end, high cost commercial load balancers including BigIP, Cisco LocalDirector, and more in serious production environments, pfSense is not nearly as powerful and flexible as enterprise-grade commercial load balancing solutions. It is not suitable for deployments that require extremely flexible monitoring and balancing configurations. For large or complex deployments, a more powerful solution is usually called for. However, the functionality available in pfSense suits countless sites very well for basic needs.

Full-featured load balancer packages are available for pfSense, such as **HAProxy** and **Varnish**, but the built-in load balancer based on `relayd` from OpenBSD does a great job for many deployments. Monitors in `relayd` can check proper HTTP response codes, check specific URLs, do an ICMP or TCP port check, even send a specific string and expect a specific response.

TCP services in the pfSense Load Balancer are handled in a *redirect* manner, meaning they work like intelligent port forwards and not like a proxy. The source address of the client is preserved when the connection is passed to internal servers, and firewall rules must allow traffic to the actual internal address of pool servers. When `relayd` is configured to handle DNS, however, it works like a proxy, accepting connections and creating new connections to internal servers.

Servers in Load Balancing pools are always utilized in a round-robin manner. For more advanced balancing techniques such as source hashing, try a reverse proxy package such as **HAProxy** instead.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the January 2015 Hangout on Server Load Balancing and Failover.

WIRELESS

Recommended Wireless Hardware

A variety of wireless cards are supported in FreeBSD 10.3-RELEASE, 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.

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.

Status of 802.11n Support

pfSense 2.3.3-RELEASE is based on FreeBSD 10.3-RELEASE 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.

Status of 802.11ac Support

Currently, there is no support for 802.11ac in FreeBSD nor in pfSense.

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.

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 the [pfSense wiki](#).

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.

mw(4)

The Marvell IEEE 802.11 wireless network driver, *mw(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 Broadcom 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.

Hardware Support Specifics

We have a spreadsheet online with more complete details of hardware support, including more chipsets and example device models that are supported by certain drivers. Currently this information is held on a public [Google Docs spreadsheet](#) linked from the [documentation wiki article on wireless support](#). As noted earlier in this chapter, often manufacturers will change device chipsets but not model numbers, so it's a rough guide at best, but it can still give some useful guidance.

Working with Virtual Access Point Wireless Interfaces

pfSense 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.

Creating and Managing Wireless Instances

To create a new wireless instance:

- Navigate to **Interfaces > (assign)** 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*.

Wireless Interface Configuration	
Parent Interface	ath0 (Atheros 9280)
Mode	Access Point
Description	Guest Wireless
A description may be entered here for administrative reference (not parsed).	

Fig. 24.1: Adding a Wireless Instance

Once the entry has been saved it is then available for assignment under **Interfaces > (assign)**. 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.

Wireless WAN

A wireless card in a firewall running pfSense can be used as the primary WAN interface or an additional WAN in a multi-WAN deployment.

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 > (assign)**
- 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 > (assign)**

- Select the wireless interface as **WAN**
- Click Save

Figure *Wireless WAN Interface Assignment* shows an Atheros card assigned as WAN.



Fig. 24.2: Wireless WAN Interface Assignment

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**

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.

WAN2 Interface (opt1, ath0)	
Status	associated
DHCP	up  Release
MAC Address	04:f0:21:0f:81:af
IPv4 Address	203.0.113.105
Subnet mask IPv4	255.255.255.0
Gateway IPv4	203.0.113.1
IPv6 Link Local	fe80::6f0:21ff:fe0f:81af%ath0_wlan0
MTU	1500
Media	MCS mode 11ng
Channel	6
SSID	GuestWireless
BSSID	12:18:d6:a9:12:ff
Rate	86M
RSSI	26.5
In/out packets	2464/2465 (67 KiB/67 KiB)
In/out packets (pass)	2464/2465 (67 KiB/67 KiB)
In/out packets (block)	10/0 (1 KiB/0 B)
In/out errors	0/4
Collisions	0

Fig. 24.3: Associated Wireless WAN Interface

WAN2 Interface (opt1, ath0)	
Status	no carrier
DHCP	down 
MAC Address	04:f0:21:0f:81:af
MTU	1500
Media	autoselect mode 11g
Channel	6
SSID	GuestWireless
In/out packets	0/3 (0 B/168 B)
In/out packets (pass)	0/3 (0 B/168 B)
In/out packets (block)	0/0 (0 B/0 B)
In/out errors	0/8
Collisions	0

Fig. 24.4: No carrier on wireless WAN

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.

Status (WAN2)						
Nearby Access Points or Ad-Hoc Peers						
SSID	BSSID	CHAN	RATE	RSSI	INT	CAPS
GuestWireless	12:18:d6:a9:12:ff	6	54M	-70:-96	100	EPS RSN HTCAP WME ATH
tardis	0e:18:d6:a9:12:ff	6	54M	-70:-96	100	EPS RSN HTCAP WME ATH

Fig. 24.5: Wireless Status

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.

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.

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.

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.

Using an External 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 replaced an existing wireless router, it could still be used to handle the wireless portion of the network if desired. 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. This way an 802.11ac wireless network may still be used and secured by pfSense at the border, even though pfSense does not yet have support for 802.11ac cards.

This technique is also commonly used with wireless equipment running *WRT, Tomato, or other custom firmware for use as dedicated access points rather than edge routers.

Turning a wireless router into an access point

When replacing a simple wireless router such as a Linksys, D-Link or other home grade device with pfSense as a perimeter firewall, the wireless functionality can be retained. 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 will handle this function for the network, and having two DHCP servers on the same broadcast domain will cause problems.

Change the LAN IP address

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 replaced this wireless router, then the wireless router was probably using the same IP address now assigned to the pfSense LAN interface, so it must be changed. A functional IP address on the access point is required for management purposes and to avoid IP address conflicts.

Plug in the LAN interface

Most wireless routers bridge their wireless network to the 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 wireless router depends on the chosen network design. The next sections cover options and considerations for selecting the best deployment style.

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 will work fine, but offers limited control over the ability of the wireless clients to communicate with internal systems. See [Choosing Routing or Bridging](#) for details on bridging in this role.

Bridging wireless to an OPT interface

For increased control over wireless clients, adding an OPT interface to the firewall for the access point is the preferred solution. To keep wireless and wired networks on the same IP subnet and broadcast domain, the OPT interface may be bridged to the LAN interface. This scenario is functionally equivalent to plugging the access point directly into the LAN switch, except pfSense 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.

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.

pfSense as an Access Point

With a wireless card that supports hostap mode (See *Cards Supporting Access Point (hostap) Mode*), pfSense can be configured as a wireless access point.

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.

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 [FreeRADIUS2 pfSense package](#) may be used for 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 [IPv4 DHCP Server](#).

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**.

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.

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.

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

Floating	WAN	LAN	CONFROOM	IPsec							
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✓	0/0 B	IPv4 ICMP echoreq	CONFROOM net	*	This Firewall	*	*	none	Allow ICMP echo (ping) for diagnostics	  
☐	✓	0/0 B	IPv4 UDP	CONFROOM net	*	CONFROOM address	500 (ISAKMP)	*	none	Allow IKE for IPsec	  
☐	✓	0/0 B	IPv4 UDP	CONFROOM net	*	CONFROOM address	4500 (IPsec NAT-T)	*	none	Allow NAT-T for IPsec	  
☐	✓	0/0 B	IPv4 ESP	CONFROOM net	*	CONFROOM address	*	*	none	Allow ESP for IPsec	  

Fig. 24.6: Rules to allow only IPsec from wireless

troubleshooting. This assumes the default UDP port 1194 is in use. If another protocol or port was chosen, adjust the rule accordingly.

Floating	WAN	LAN	CONFROOM	OpenVPN							
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✓	0/0 B	IPv4 ICMP echoreq	CONFROOM net	*	This Firewall	*	*	none	Allow ICMP echo (ping) for diagnostics	   
☐	✓	0/0 B	IPv4 UDP	CONFROOM net	*	CONFROOM address	1194 (OpenVPN)	*	none	Allow OpenVPN to the firewall	   

Fig. 24.7: Rules to allow only OpenVPN from wireless

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.

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.

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.

Floating	WAN	LAN	EXTERNALAP	IPsec	OpenVPN						
Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✔	0/0 B	IPv4 ICMP echo req	EXTERNALAP net	*	EXTERNALAP address	*	*	none	Allow ping to the firewall	
☐	✔	0/0 B	IPv4 TCP/UDP	EXTERNALAP net	*	*	53 (DNS)	*	none	Allow DNS – will be trapped by NAT rule	
☐	✖	0/0 B	IPv4 *	*	*	RFC1918	*	*	none	Deny access to any other private networks	
☐	✔	0/0 B	IPv4 *	EXTERNALAP net	*	*	*	*	none	Allow Internet access	

Fig. 24.8: Rules to allow only Internet access from wireless

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.

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.

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.

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](#) may be found on the wiki, including access point capability descriptions.

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.

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.

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.

Interface Unavailable for Assignment

If a wireless interface does not appear in the list of interfaces **Interfaces > (assign)** there are two possible 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.

pfSense 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 book. For those seeking such information, we recommend the book *[802.11 Wireless Networks: The Definitive Guide](#)*.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the May 2015 Hangout on Wireless Access Points.

CAPTIVE PORTAL

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.

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.

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.

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](#).

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*.

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.

Zone Configuration Options

This section describes each of the Captive Portal configuration options. These options are available once a Captive Portal zone has been created under **Services > Captive Portal**. These options all work independently of one another for each zone. For example, the allowed IP addresses specified in a zone only affect that one zone.

Interface Determines the interfaces that will be active for 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 would leave the firewall unresponsive.

Idle timeout A timeout, specified in minutes, after which idle users will be disconnected. Users may log back in immediately.

Hard timeout A timeout, specified in minutes, which will forcefully log off users after a specified period. Either a hard timeout, idle timeout or both should be entered to ensure sessions are removed if users do not log off, as most users likely will not. Users will be able to 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.

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 need to login. By setting the hard timeout to a value such as 1 hour, the client would effectively be limited to three hours of access before needing 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 used up, the client can only log in with valid credentials until the waiting period specified below has expired.

Waiting period to restore pass-through credits The number of hours after which clients will have their available pass-through credits restored to the original count after using the first one. This must be above 0 hours if pass-through credits are enabled.

Reset waiting period on attempted access If enabled, the waiting period is reset to the original duration if access is attempted when all pass-through credits have already been exhausted. This will prevent people who repeatedly attempt to access the portal from gaining open access too quickly.

Logout popup window When checked, a logout pop up window is shown to the user which allows clients to explicitly disconnect themselves before the idle or hard timeout occurs. Unfortunately, since most browsers have pop up blockers enabled, this window may not work for most clients unless the portal can be excluded from pop-up blocking in their browsers.

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`.

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 in order to properly handle this redirect.

```
<?php
require("globals.inc");
$request_uri = urldecode(str_replace("/index.php?redirurl=", "", $_SERVER["REQUEST_URI"]));
$portal_redirurl = urldecode("$PORTAL_REDIRURL$");
if(!strstr(urldecode("$PORTAL_REDIRURL$"), $request_uri)) {
    Header("Location: $PORTAL_REDIRURL$");
    exit;
}
?>
```

After authentication Redirection URL After authenticating or clicking through the portal, users will be redirected to this URL rather than the one they originally tried to access. If this field is left blank, the user will be redirected to the URL the user initially attempted to access.

Concurrent user logins When checked, only one login per user account is allowed. The most recent login is permitted and any previous logins under that username will be disconnected. This is not a total limit for the entire portal, but a per-account limit.

MAC filtering When set, MAC address filtering is disabled. 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 router's MAC address. If this option is set, no attempt will be made to ensure that the MAC address of clients stay the same while they're logged into the portal.

Note: If this option is checked is enabled, RADIUS MAC authentication cannot be used.

Pass-through MAC Auto Entry In some use cases, users may only need to authenticate once per device, and then never 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, a MAC passthrough entry is automatically added after the user has successfully authenticated. Users of that MAC address will never have to authenticate again unless the entry is manually removed. To remove the passthrough MAC entry, log in and remove it manually from the Pass-through MAC tab.

Note: If this option is enabled, RADIUS MAC authentication cannot be used and the logout window will not be shown.

Pass-through MAC automatic addition with username If this option is set, the username used during authentication will be saved 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 may have unlimited bandwidth.

Authentication

This section allows authentication to be configured. If authentication is required, either the local user manager or RADIUS authentication may be used.

No Authentication When selected, users need only click through the portal screen for access. The form must still be submitted, but it does not need to have any user entry fields, only a submit button.

Local User Manager / Vouchers This option allows users to authenticate with a username and password, but not with a RADIUS server. Captive Portal users in this mode are managed in the pfSense GUI. Local users are added in the User Manager [User Management and Authentication](#).

Additionally, if **Allow only users/groups with ‘Captive portal login’ privilege set** is checked, then to access the portal, users must have the Captive Portal privilege on their account, or be a member of a group containing this privilege.

Vouchers are pre-generated access codes that can be provided to users to grant short-term access. Vouchers may be used in addition to, or instead of, local user authentication. For more information on using vouchers, see [Vouchers](#) later in this chapter.

RADIUS Authentication When selected, the RADIUS server options are displayed and Captive Portal users in this zone will be validated against the configured RADIUS server(s).

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. 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 [RADIUS Authentication with Windows Server](#). 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.

To use RADIUS, select **RADIUS Authentication** under **Authentication** and then fill in the data about the RADIUS server.

RADIUS Protocol Control which protocol the RADIUS server can use for authentication. The available choices are:

PAP (Password Authentication Protocol) The least secure but most compatible option, PAP sends passwords in plain text.

CHAP_MD5 (Challenge-Handshake Authentication Protocol) More secure than PAP, CHAP uses MD5 and encrypts the password during transmission. While more secure than PAP on the wire, the server side must know the cleartext password in order to calculate the challenge.

MSCHAPv1 (Microsoft CHAP, Version 1) A Microsoft-designed variation of CHAP primarily used in older versions of Windows (NT 3.x through Windows 95). There are programs available that can easily capture the password hashes from the exchange.

MSCHAPv2 (Microsoft CHAP, Version 2) Adds more security features on top of CHAP/MS-CHAP v1.

The relative (in)security of these protocols may be of little consequence depending on the layout of the network and the location of the RADIUS server, but should still be considered.

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.

Primary Authentication Source

The Primary/Secondary RADIUS 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 primary RADIUS server is down, the secondary RADIUS server will be tried.

IP Address The IP address or hostname of the RADIUS server

Port The authentication port for the RADIUS server, typically 1812.

Shared Secret The shared secret for this firewall on the RADIUS server.

Secondary Authentication Source

The secondary authentication source defines a completely separate RADIUS authentication setup from the primary. For example, the primary RADIUS source could be traditional usernames and passwords, while the secondary could be pre-paid card numbers or PINs. As with the primary authentication source, a primary server and secondary server may be defined.

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>
```

Accounting

RADIUS accounting will send session information back to the RADIUS server indicating when a user's session starts, ends, and how much data they have transmitted.

Warning: Not all RADIUS servers support or are configured to accept accounting data, so make sure that the RADIUS server has been setup up properly before enabling this feature.

Accounting Port Configures the port upon which the RADIUS server accepts accounting packets, typically 1813.

Accounting Updates This configures what specific type of accounting is supported by the server.

No updates Synonymous with disabling accounting, it will not send accounting updates to the server.

Stop/start Will send START and STOP records for a user session only.

Stop/start (FreeRADIUS) Will send START and STOP records for a user session only, in a way that is compatible with FreeRADIUS.

Interim Will send START and STOP records and also periodically send updates to the server while a user session is active. This 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.

RADIUS Options

These options fine-tune how RADIUS authentication behaves.

Reauthentication If enabled, `Access-Request` packets will be sent 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. Note 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.

RADIUS MAC authentication If this option is enabled, the captive portal will attempt to authenticate users by sending their MAC address as the username and the password entered into **MAC authentication secret** to the RADIUS server. This option cannot be used if MAC filtering is disabled.

RADIUS NAS IP attribute This field controls what is sent to the RADIUS server in the `Calling-Station` attribute. Choose the Interface/IP address to use from the drop-down list.

Session-Timeout When **Use RADIUS Session-Timeout attributes** is enabled, clients will be disconnected after the amount of time retrieved from the `RADIUS Session-Timeout` attribute

Type Sets the RADIUS vendor type for the client behavior. If RADIUS Type is set to *Cisco*, Access-Request packets will have the value of Calling-Station-Id set to the client IP address and the Called-Station-Id set to the client MAC address. *Default* behavior is Calling-Station-Id = client MAC address and Called-Station-Id = firewall WAN IP address.

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.

NAS Identifier The hostname of the firewall is sent as the NAS Identifier by default. Specify a **NAS Identifier** here to override the default value

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

HTTPS login

Check this box to use HTTPS for the portal page. If enabled, an SSL Certificate must also be selected.

HTTPS server name Specify the FQDN (hostname + domain) to be used for HTTPS. This must match the Common Name (CN) on the certificate to prevent users from receiving certificate errors in their browsers.

SSL Certificate Select the SSL certificate to be used by the portal for HTTPS logins. Certificates are managed in [Certificate Management](#).

Disable HTTPS Forwards When checked, attempts 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.

HTML Page Contents

Using these controls, custom HTML pages can be uploaded 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: Because custom portal pages can run PHP, ensure to properly secure the code in the page so it 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 using 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**
- 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 25.1: Download: example-noauth.html

```

1 <html>
2 <head>
3   <title>Welcome to our portal</title>
4 </head>
5 <body>
6   <p>Welcome to our portal</p>
7   <p>Click Continue to access the Internet</p>
8   <form method="post" action="$PORTAL_ACTION$"
9     <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$"
10    <input name="zone" type="hidden" value="$PORTAL_ZONE$"
11    <input name="accept" type="submit" value="Continue">
12  </form>
13 </body>
14 </html>

```

Portal page with authentication Here is an example portal page requiring authentication.

Listing 25.2: Download: example-auth.html

```

1 <html>
2 <head>
3   <title>Welcome to our portal</title>
4 </head>
5 <body>
6   <p>Welcome to our portal</p>
7   <p>Enter your username and password and click Login to access the Internet</p>
8   <form method="post" action="$PORTAL_ACTION$"
9     <input name="auth_user" type="text">
10    <input name="auth_pass" type="password">
11    <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$"
12    <input name="zone" type="hidden" value="$PORTAL_ZONE$"
13    <input name="accept" type="submit" value="Login">
14  </form>
15 </body>
16 </html>

```

Portal page with Vouchers Here is an example portal page for use with vouchers.

Listing 25.3: Download: example-voucher.html

```

1 <html>
2 <head>
3     <title>Welcome to our portal</title>
4 </head>
5 <body>
6     <p>Welcome to our portal</p>
7     <p>Enter your voucher code and click Login to access the Internet</p>
8     <form method="post" action="$PORTAL_ACTION$">
9         <input name="auth_voucher" type="text">
10        <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
11        <input name="zone" type="hidden" value="$PORTAL_ZONE$">
12        <input name="accept" type="submit" value="Login">
13    </form>
14 </body>
15 </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:

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.

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.

Listing 25.4: Download: example-logout.html

```

1 <html>
2 <head><title>Redirecting...</title></head>
3 <body>
4 <span style="font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif; font-size: 11px;">
5 <b>Redirecting to <a href="<?=$my_redirurl;?>"><?=$my_redirurl;?></a>...</b>
6 </span>
7 <script type="text/javascript">
8 //
9 LogoutWin = window.open('', 'Logout', 'toolbar=0,scrollbars=0,location=0,statusbar=0,menubar=0,resizable=0,width=300,height=300');
10 if (LogoutWin) {
11     LogoutWin.document.write('&lt;html&gt;');
12     LogoutWin.document.write('&lt;head&gt;&lt;title&gt;Logout&lt;/title&gt;&lt;/head&gt;');
13     LogoutWin.document.write('&lt;body style="background-color:#435370"&gt;');
14     LogoutWin.document.write('&lt;div class="text-center" style="color: #ffffff; font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif; font-size: 11px;"&gt;');
15     LogoutWin.document.write('&lt;b&gt;Click the button below to disconnect&lt;/b&gt;&lt;p /&gt;');
16     LogoutWin.document.write('&lt;form method="POST" action="&lt;?=$logouturl;?&gt;"&gt;');
17     LogoutWin.document.write('&lt;input name="logout_id" type="hidden" value="&lt;?=$sessionid;?&gt;" /&gt;');
18     LogoutWin.document.write('&lt;input name="zone" type="hidden" value="&lt;?=$cpzone;?&gt;" /&gt;');
19     LogoutWin.document.write('&lt;input name="logout" type="submit" value="Logout" /&gt;');
20     LogoutWin.document.write('&lt;/form&gt;');
21     LogoutWin.document.write('&lt;/div&gt;&lt;/body&gt;');
22     LogoutWin.document.write('&lt;/html&gt;');
23     LogoutWin.document.close();
24 }
25
26 document.location.href="&lt;?=$my_redirurl;?&gt;";
27 //]]&gt;
28 &lt;/script&gt;
29 &lt;/body&gt;
30 &lt;/html&gt;
</pre>
</div>
<div data-bbox="111 930 333 947" data-label="Page-Footer">25.4. MAC Address Control</div>
<div data-bbox="849 930 889 947" data-label="Page-Footer">517</div>
```

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 .

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.

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.

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 commonly implemented in places where authenticated, but time-limited, Internet access is desired without needing to provide a username and password to users.

This type of configuration is frequently found in places such as coffee shops, hotels, and airports. Users enter a voucher code in the portal login form and are granted access for the amount of time configured for the voucher roll. Voucher rolls can be exported as a CSV file, and some companies have even integrated the exported voucher lists into their point of sale applications to print a voucher 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 from the same or a different computer. If the voucher is used again from another computer, the previous session is stopped.

A public/private RSA key pair is required to generate and verify vouchers. A 32-bit set of keys is generated automatically by the firewall the first time the page is visited, and a custom key pair may be generated manually if desired. The maximum key length supported is 64 Bits. Using shorter keys will make the generated vouchers shorter but eventually less secure.

The voucher settings are unique per Captive Portal Zone. To configure vouchers:

- Navigate to **Services > Captive Portal**
- Click  on the line for the Zone to edit
- Click the **Vouchers** tab
- Check **Enable**
- Fill in the form as needed. In most cases, the remaining options on this screen can be left at their default values.

Voucher Rolls Voucher rolls are managed in this section of the screen. Information about each roll is listed along with a link to Add new rolls. No options appear here until after the other settings have been configured and saved. See [Managing Voucher Rolls](#) for more.

Voucher Keys The keys used to generate and verify vouchers. Before vouchers are enabled on a zone, these values are randomly generated each time the page loads. Once vouchers are enabled and saved, the keys are set.

Warning: Take care not to change the keys or other bits once they have been saved, or all current vouchers are rendered invalid and new voucher rolls must be created.

Voucher Public Key This key is used to decrypt vouchers. The pre-generated key may be used, or click **Generate new keys** to make a new public and private key pair. A key may also be generated elsewhere and then used by pasting the RSA public key (64 Bit or smaller) in PEM format here.

Voucher Private Key This key is used to generate voucher codes and does not need to be available if the vouchers have been generated offline. The pre-generated key may be used, or paste in an RSA private key (64 Bit or smaller) in PEM format here if the key was generated elsewhere.

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. Using a smaller character set will generally cause longer vouchers to ensure randomness.

Voucher Bits The following “bit” fields control how the vouchers themselves are generated. Leaving these values at their defaults is recommended but they may be adjusted if required. 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 used to store 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 used to store 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 stored in every voucher, and is verified during voucher check. The size of the magic number depends on how many bits are left by adding together the number of bits for the roll, ticket, and checksum. If all bits are used, no magic number will be used or checked.

Invalid Voucher Message This message is displayed 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 if they enter a voucher that was valid, but has expired.

- Click **Save**

Once the settings have been saved, the voucher management controls will be active.

Managing Voucher Rolls

Vouchers are created 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 their code was picked from.

Creating Voucher Rolls

To create a roll, click  **Add** under the roll list.

Roll # The number of this roll. Each roll must have a unique number. This can be any number from 0 to 65535.

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 how many vouchers will be made on this roll. The value can be from 0 to 1023.

Note: If the count on an existing roll is changed, it will invalidate **all** other vouchers on the roll, so it is best not to change this once a roll has been created.

Comment A description of the roll for reference, such as 2 hour vouchers for coffee purchases.

Click **Save** and the new roll is ready to use.

Editing Existing Rolls

An existing voucher roll can be edited by clicking , but be careful when making changes. Changing the **Roll** number or **Count** will cause the current vouchers on the roll to be invalidated.

Removing Voucher Rolls

Rolls of vouchers can be removed by clicking  at the end of their row. When a roll is removed, *all* of the vouchers in that roll become invalid, so do not remove a roll unless it has been completely used, compromised, or otherwise is no longer needed.

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.

This file can be saved and opened in nearly any spreadsheet editor such as LibreOffice Calc, Google Docs, or Excel. From there they can be printed, fed into a POS system, and so on.

Using Vouchers on A Portal Page

Vouchers must be submitted via the `auth_voucher` form field. See *Portal page with Vouchers* for an example.

Viewing Active Vouchers

A list of currently active vouchers and their timers can be found at **Status > Captive Portal**, on the **Active Vouchers** tab for a zone, as seen in Figure *Active Vouchers*.

Viewing Voucher Roll Utilization

A list of voucher rolls and how many have been used can be found at **Status > Captive Portal**, on the **Voucher Rolls** tab for a zone, as in Figure *Vouchers Roll Usage*

Active Users	Active Vouchers	Voucher Rolls	Test Vouchers	Expire Vouchers
Vouchers in Use (1)				
Voucher	Roll	Activated at	Expires in	Expires at
GwQUjGc24y43	1	06/13/2016 10:23:01	58min	06/13/2016 11:23:01

Fig. 25.1: Active Vouchers

Active Users	Active Vouchers	Voucher Rolls	Test Vouchers	Expire Vouchers			
Roll#	Minutes/Ticket		# of Tickets	Comment	used	active	ready
1	60		1000		232	1	767

Fig. 25.2: Vouchers Roll Usage

Testing Vouchers

A voucher code can be tested for validity by entering it on **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.

QmSCLG6Wre63 (1/278) good for 60 Minutes
Access granted for 60 Minutes in total.

Active Users	Active Vouchers	Voucher Rolls	Test Vouchers	Expire Vouchers

Test Vouchers

Vouchers

Enter multiple vouchers separated by space or newline. The remaining time, if valid, will be shown for each voucher.

Test

Fig. 25.3: Testing Vouchers

Expiring Vouchers

Vouchers can be invalidated during or before use by entering them at **Status > Captive Portal**, on the **Expire Vouchers** tab for a zone. After submitting, any voucher listed in the form will no longer work. Active vouchers entered on this page are also immediately expired.

Any number of vouchers may be expired in a batch. Enter vouchers separated by newlines.

Synchronizing Vouchers

At the bottom of the **Vouchers** tab there are options to synchronize vouchers to another unit. This works similarly to the XML-RPC configuration synchronization found in high availability setups (See [pfSense XML-RPC Config Sync Overview](#)). When configured, this will copy the voucher rolls to the target unit and also push information about active vouchers to the target unit as the vouchers are used.

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 be `admin`).

Voucher sync password The GUI password for the target system.

Unlike the configuration synchronization in a high availability cluster, this synchronization is configured on both the master and slave nodes. This is done to ensure that vouchers used on the slave node while it is active are also sent back to the master when it returns to an active state. Unlike the configuration synchronization, this does not create a loop.

Manually Generating RSA Keys for Vouchers

RSA public and private keys can be manually created on a separate system for use with vouchers. The commands to generate a 64-bit key for the vouchers are:

```
$ openssl genrsa 64 > key64.private
$ openssl rsa -pubout < key64.private >key64.public
```

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**.

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:

```

```

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>
```

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.

Viewing Authenticated Captive Portal Users

A list of currently active users for the portal can be found under **Status > Captive Portal**. First, pick a zone and then a list of online users for that zone is displayed. Depending on the authentication settings, 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.

Active Users	Active Vouchers	Voucher Rolls	Test Vouchers	Expire Vouchers

Captive Portal Zone				
Display Zone	MyZone			

Users Logged In (1)				
IP address	MAC address	Username	Session start	Actions
10.7.0.10	00:0c:29:4cb3:9b	unauthenticated	06/13/2016 12:27:09	

 Show Last Activity

Fig. 25.4: 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.

Users Logged In (1)				
IP address	MAC address	Username	Session start	Actions
10.7.0.10	00:0c:29:4c:b3:9b	jimp	06/13/2016 12:29:51	

Fig. 25.5: 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*.

Users Logged In (1)				
IP address	MAC address	Username	Session start	Actions
10.7.0.10	00:0c:29:4c:b3:9b	dCKPsPLLiL83	06/13/2016 12:31:23	

Fig. 25.6: Online Captive Portal Users Vouchers

Troubleshooting Captive Portal

This section contains troubleshooting tips for the most common problem with captive portal.

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.

Portal Page never loads (times out) nor will any other page load

This is most commonly a DNS failure. If the firewall is not used for DNS, and there are no entries under the **Allowed IP Address** tab (*Allowed IP Address*) for the DNS servers, then the users cannot contact DNS to resolve a hostname. If they cannot resolve a hostname, then their browser will never even attempt to load a web page, and thus will never be redirected to the portal page.

To resolve this, either use the firewall IP address and the DNS forwarder for client DNS, or add **Allowed IP Address** entries for the external DNS servers.

Another possible explanation for this is that the firewall rules on the interface with the portal do not allow the users to reach web sites on port 80. Ensure the firewall rules pass out traffic to TCP port 80, and to make sure DNS works, they must also be able to reach the configured DNS servers on TCP and UDP port 53.

HTTPS home page users do not get redirected to the portal page

It is not possible to redirect HTTPS browsing attempts in a way that will work for users securely and without error. It would either not work at all, or give the user a scary SSL certificate warning for a site they usually trust. Direct users to load an HTTP site and then they will be redirected to the portal and receive a login prompt.

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.

Port forwards to hosts behind the portal only work when the target system is logged 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.

The Captive Portal feature of pfSense redirects users to a web page hosted on the firewall before Internet access is permitted. From this page, users can be forced to authenticate before access is granted, or perform a simple click-through. The most common uses of Captive Portal are for wireless hot spots, or for additional authentication before allowing access to internal networks from wireless clients. It can also be used with wired clients if desired. Captive Portal is configured under **Services > Captive Portal**.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) to view the April 2015 Hangout on Captive Portal

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.

Does not yet support IPv6

Currently, Captive Portal does not support IPv6.

Not capable of reverse portal

A reverse portal, requiring authentication for traffic coming into a local network from the Internet, is not possible.

HIGH AVAILABILITY

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.

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.

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.

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.

pfSense XML-RPC Config Sync Overview

To make the job of maintaining practically identical firewall 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 have the GUI running on the same port and protocol, for example: HTTPS on port 443, which is the default setting. The admin account cannot be disabled and both nodes must have the same admin account password.

Example Redundant Configuration

This section 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 pfsync interface being used solely 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.

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 26.1: 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 26.2: 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 26.3: 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](#).

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.

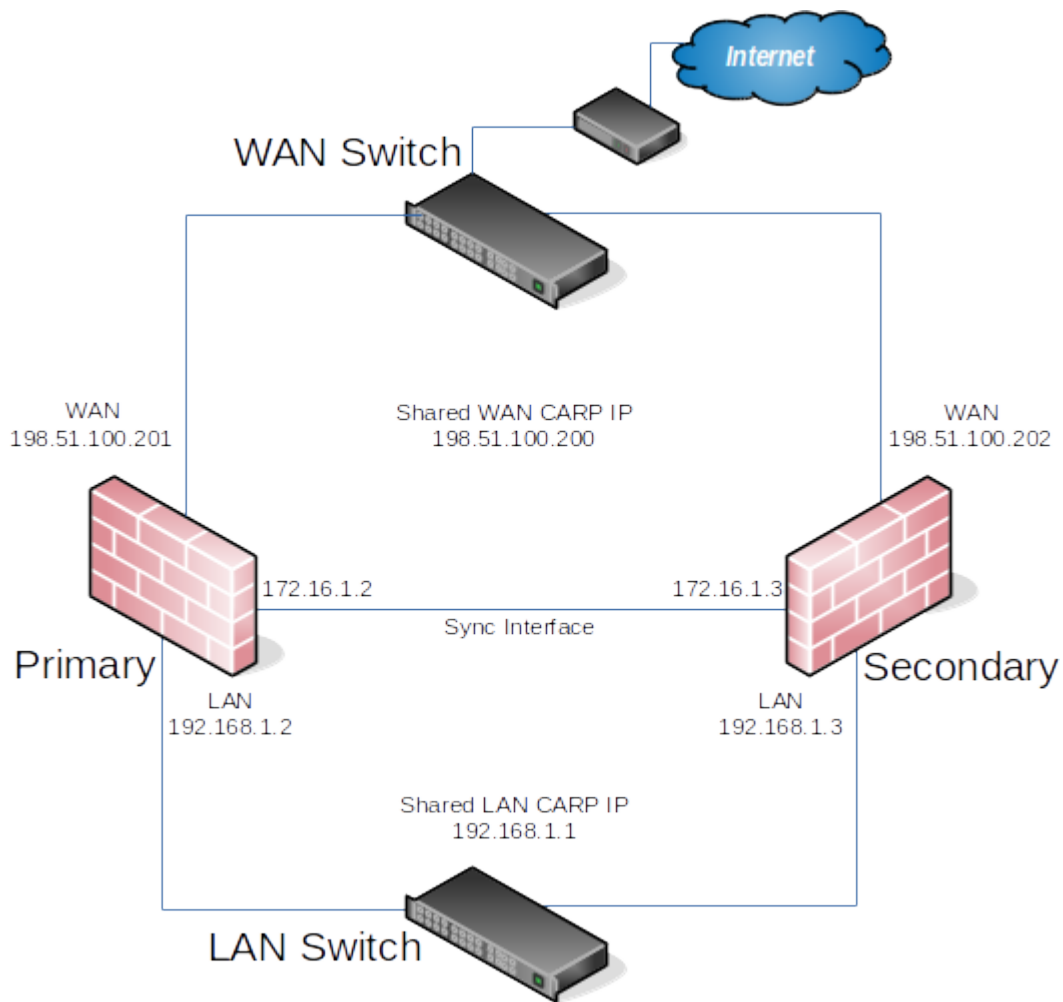


Fig. 26.1: Example HA Network Diagram

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.

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. 26.2: 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.

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**

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 always be `admin`, no other user will work!

- 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. 26.3: 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. 26.4: 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.

Multi-WAN with HA

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.

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 26.4: 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 26.5: 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 26.6: 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 26.7: 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 26.8: Sync IP Address Assignments

IP Address	Usage
172.16.1.2	Primary firewall Sync
172.16.1.3	Secondary firewall Sync

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.

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.

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*.

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 [High Availability Troubleshooting](#).

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.

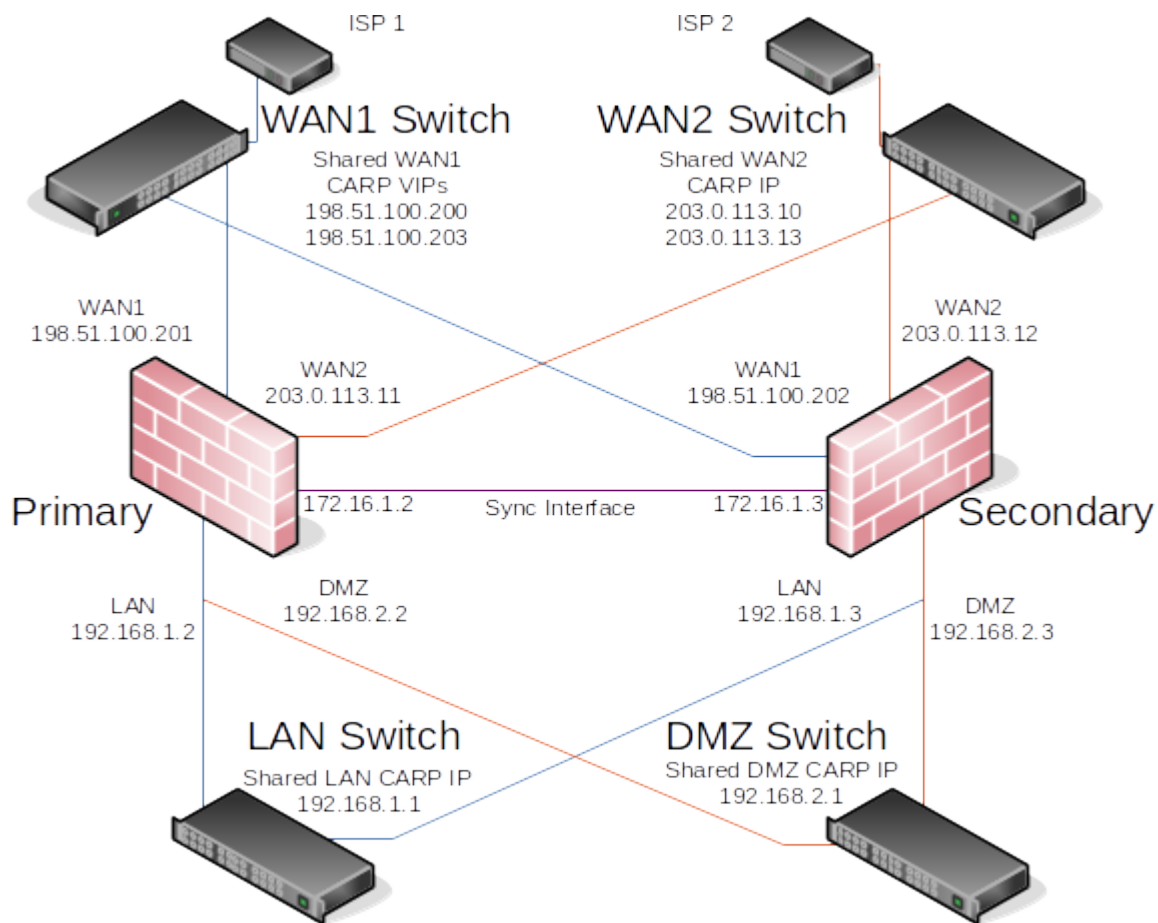


Fig. 26.5: Diagram of Multi-WAN HA with DMZ

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.

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. 26.6: DHCP Failover Pool Status

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

Providing Redundancy 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.

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.

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.

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.

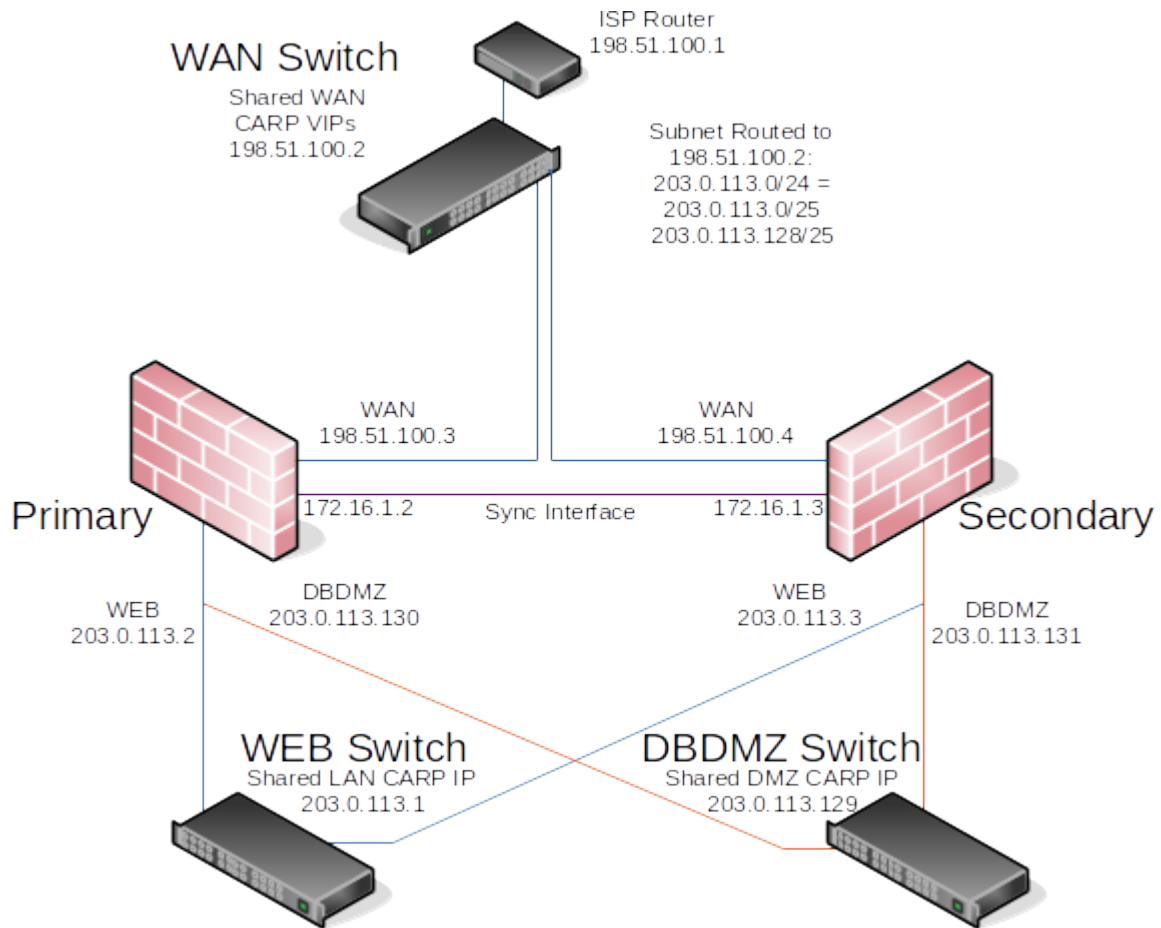


Fig. 26.7: Diagram of HA with Routed IPs

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.

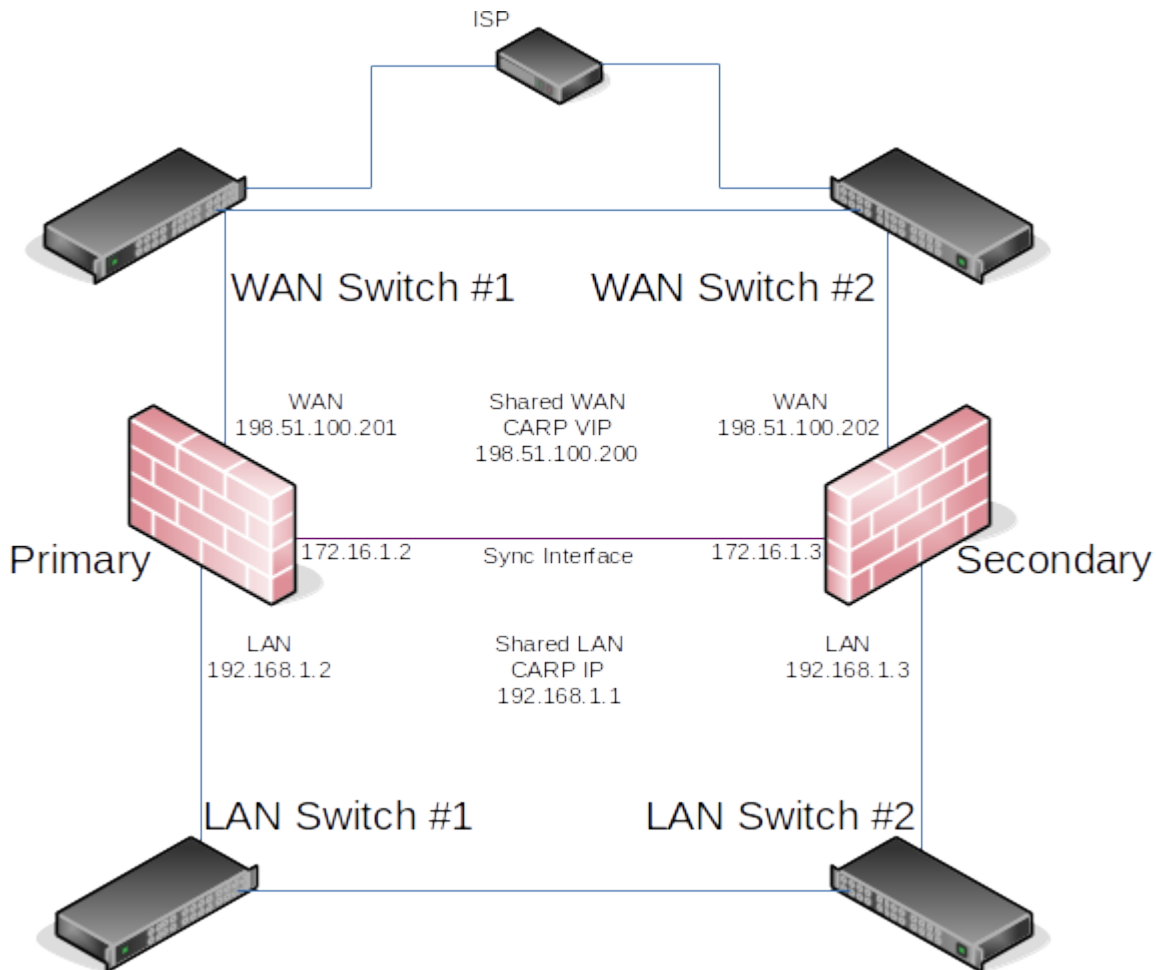


Fig. 26.8: Diagram of HA with Redundant Switches

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.

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 book.

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.

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](#).

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.

Interface

If multiple subnets are required on a single interface with HA, this may be accomplished using IP Aliases. As with the main interface IP addresses, we recommend each firewall have an IP address inside the additional subnet, for a total of at least three IPs per subnet. Separate IP alias entries must be added to each node inside the new subnet, ensuring that their subnet masks match the actual subnet mask for the new subnet. IP alias VIPs that are directly on an interface do not sync, so this is safe.

Once the IP Alias VIP has been added to both nodes to gain a foothold in the new subnet, CARP VIPs may then be added using IP addresses from the new subnet.

It is possible to omit the IP Aliases and use a CARP VIP directly in the other subnet so long as communication between the additional subnet and both individual HA nodes is not required.

High Availability Troubleshooting

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 [CARP/VIPs board on the pfSense 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.

Common Misconfigurations

There are three common misconfigurations that happen which prevent HA from working properly.

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. 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 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.

IP Address for CARP Interface

The interface upon which the CARP VIP resides must already have another IP defined directly on the interface (VLAN, LAN, WAN, OPT) before it can be utilized.

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).

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 [Issues inside of Virtual Machines \(ESX\)](#).

Primary system is stuck as BACKUP

In some cases, this 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**.

From the shell or **Diagnostics > Command**, run the following command to check for a demotion:

CARP has detected a problem and this unit has been demoted to BACKUP status.
Check the link status on all interfaces with configured CARP VIPs.
Search the [System Log](#) for CARP demotion-related events.

[Reset CARP Demotion Status.](#)

[Temporarily Disable CARP](#) [Enter Persistent CARP Maintenance Mode](#)

CARP Interfaces		
CARP Interface	Virtual IP	Status
WAN@200	198.51.100.200/24	BACKUP
LAN@1	192.168.1.1/24	INIT

Fig. 26.9: CARP Status when Primary is demoted

```
# 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.

Issues inside of Virtual Machines (ESX)

When using HA inside of a Virtual Machine, especially VMware ESX, some special configurations are needed:

1. Enable promiscuous mode on the vSwitch.
2. Enable “MAC Address changes”.
3. Enable “Forged transmits”.

ESX VDS Promiscuous Mode 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 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) was used in 4.0 or 4.1 and upgrade from 4.0 to 4.1 or 5.0, the VDS will not properly pass CARP traffic. If a new VDS was 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 and then re-enable promiscuous mode.

ESX Client Port Issues

If a physical HA cluster is connected to a switch with an ESX host using multiple ports on the ESX host (lagg group or similar), and only certain devices/IPs are reachable by the target VM, then the port group settings may need adjusting in ESX to set the load balancing for the group to hash based on IP, not the originating interface.

Side effects of having that setting incorrectly include:

- Traffic only reaching the target VM in promiscuous mode on its NIC.
- Inability to reach the CARP VIP from the target VM when the “real” IP address of the primary firewall can be reached.
- Port forwards or other inbound connections to the target VM work from some IP addresses and not others.

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.

KVM+Qemu Issues

Use e1000 NICs (*em(4)*), not the *ed(4)* NICs or CARP VIPs will never leave init state.

VirtualBox Issues

Setting “Promiscuous mode: Allow All” on the relevant interfaces of the VM allows CARP to function on any interface type (Bridged, Host- Only, Internal)

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.

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.

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](#)

pfSense 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, units act as an “active/passive” cluster with the primary node working as the master unit 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 pfSense 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:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) which contains the June 2015 Hangout also covering High Availability.

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.

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.

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.

SERVICES

IPv4 DHCP Server

The IPv4 **DHCP server** assigns IPv4 addresses and related configuration options to client PCs on a network. It is enabled by default on the LAN interface with a default range of 192.168.1.100 through 192.168.1.199. In this default configuration, the firewall assigns its LAN IP address (192.168.1.1) as the gateway and DNS server if either the DNS Resolver or DNS Forwarder is enabled. There are numerous options available in the web interface, which are covered in the next section.

Configuration

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.

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.

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.

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.

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.

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 is 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.

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.

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

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.

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.

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.

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.

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.

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`.

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.

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.

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. 27.1: DHCP Daemon Service Status

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 firewall's 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  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.

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.

IPv6 DHCP Server and 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**.

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.

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.

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 2¹⁶ (65,536) IPs, which is extremely large by today’s IPv4 standards, but only a small portion of the whole /64.

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:FF00:: 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::.

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.

DHCP & 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.

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).

DNS Resolver Advanced Options

pfSense 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:

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.

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.

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 [pfSense-initiated Traffic and IPsec](#) 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.

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.

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"
```

DNS Forwarder

The DNS Forwarder in pfSense 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.

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.

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 be 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.

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**.

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/
```

Dynamic DNS

The Dynamic DNS client built into pfSense 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.

Dynamic DNS and IPv6

As of this writing, there are very few Dynamic DNS providers that offer IPv6 support. The available choices are limited to HE.net when they host DNS for a domain, custom types, and RFC 2136 servers.

Configuring a Dynamic DNS Client

pfSense 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.

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.

Configuring the server infrastructure for RFC 2136 Dynamic DNS hosting is beyond the scope of this book, but there is a basic how-to on the pfSense documentation wiki that covers [setting up BIND to handle RFC 2136 updates](#).

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.

SNMP

The [Simple Network Management Protocol](#) (SNMP) daemon enables remote monitoring of some pfSense system 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 used by pfSense 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 book, 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](#).

SNMP and IPv6

The `bsnmpd` daemon does not currently support IPv6.

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.

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.

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 known 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.

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.

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.

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.

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.

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
```


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**.

UPnP & NAT-PMP Rules				
Port	Protocol	Internal IP	Int. Port	Description
51412	tcp	10.3.0.14	51412	NAT-PMP 51412 tcp
54493	tcp	10.3.0.14	54493	Transmission at 54493
54493	udp	10.3.0.14	54493	Transmission at 54493

Clear all sessions

Fig. 27.2: UPnP & NAT-PMP Status Screen Showing Client PCs With Forwarded Ports

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.

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.

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 `ntpd` 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 `ntpd` 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 `ntpd`.

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

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. 27.3: 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.

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.

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 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.

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.

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.

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.

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.

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**

PPPoE Server

pfSense 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 [RADIUS Authentication with Windows Server](#) 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**

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, which will be covered later in the book.

SYSTEM MONITORING

System Logs

pfSense 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 full installations where logs are kept on disk, they are retained across reboots. For NanoBSD installations or when `/var` is in a RAM disk, the logs reset at boot time.

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.

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

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. 28.1: Example System Log Entries

- 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**.

Changing Log Settings

Log settings may be adjusted in two different ways. First, the options can be set globally at **Status > System Logs** on the **Settings** tab. Second, each log tab can have its own unique settings which 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:

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) The size of the `clog` 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.

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 in pfSense. Setting this option will generate a large amount of log data for connections outbound from the firewall. We only recommend enabling 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.

IGMP Proxy Toggles the verbosity of the IGMP proxy logs. By default, the logs do not contain much information. Enabling this option causes IGMP proxy to log more detail.

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 The current default. 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.

Display as column 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. We do not recommend using this option as having local logs is vital for the vast majority of use cases.

Reset Log Files This button will clear the data from all log files and reinitialize them as new, empty logs. This must be done after changing the log file sizes, 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](#).

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)**, 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.

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 on the firewall itself are of a finite size and they are cleared on reboot on NanoBSD. Copying these entries to a syslog server can aid troubleshooting and enable 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.

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 follows:

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 an 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 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 pfSense 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.

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 on the documentation wiki article on the [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)

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.

Server Load Balancer Events Messages from relayd which handles server load balancing.

Network Time Protocol Events Messages from the NTP daemon and client.

Wireless Events Messages from the Wireless AP daemon, hostapd.

- Click Save to store the changes.

If a syslog server is not already available, it is fairly easy to set one up. See [Syslog Server on Windows with Kiwi Syslog](#) for information on setting up Kiwi Syslog on Windows. 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.

Configuring a Syslog Server on FreeBSD

Setting up a syslog server on a FreeBSD server requires only a couple steps. In this example, replace 192.168.1.1 with the IP address of the firewall, replace exco-rtr with the hostname of the firewall, and replace exco-rtr.example.com with the full hostname and domain of the firewall. This example uses 192.168.1.1 because the best practice is to send syslog messages using the *internal* address of a firewall, not a WAN interface.

Note: These changes must all be made on the syslog server, **not** on the firewall.

First, the firewall will likely need an entry in /etc/hosts containing the address and name of the firewall:

```
192.168.1.1          exco-rtr      exco-rtr.example.com
```

Then adjust the startup flags for syslogd to accept syslog messages from the firewall. Edit /etc/rc.conf and add this line if it does not exist, or add this option to the existing line for the setting:

```
syslogd_flags=" -a 192.168.1.1"
```

Lastly, add lines to /etc/syslog.conf to catch log entries from this host. Underneath any other existing entries, add the following lines:

```
!*
+*
+exco-rtr
*.*                               /var/log/exco-rtr.log
```


Those lines will reset the program and host filters, then set a host filter for this firewall using the short name as entered in `/etc/hosts`.

Tip: Look at `/etc/syslog.conf` on the pfSense firewall for ideas about filtering the logs for various services into separate log files on the syslog server.

After these changes, `syslogd` must be restarted. On FreeBSD this is one simple command:

```
# service syslogd restart
```

Now look at the log file on the syslog server and if the configuration is correct, it will be populating the logs with entries as activity happens on the firewall.

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. 28.2: 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.

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.

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

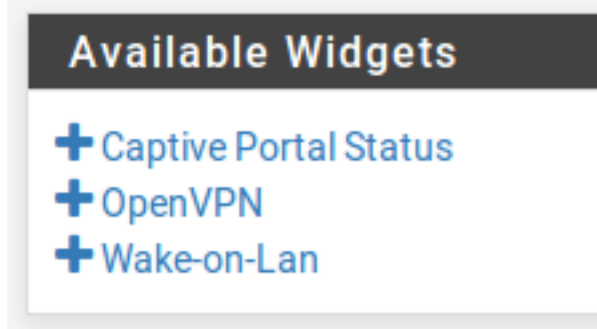


Fig. 28.3: Available Widgets List



Fig. 28.4: Widget Title Bar

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 user-name.

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 (full installs only). This automatic update check can be disabled in the update settings.

Platform The platform indicates which variation of pfSense is running. A full install will show pfSense, an embedded install shows NanoBSD.

NanoBSD boot slice If this is an embedded install, the running slice is also displayed (*pfsense0* or *pfsense1*), along with the slice that will be used for the next boot.

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 side 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 such as ALIX 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.

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 *Speed and Duplex* for more about setting an interface's speed and duplex.

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
- Click  to stop a running service
- Click  to start a stopped service

If available, other shortcuts are shown which navigate to a service configuration (), detailed status page (), or logs (). See *Quickly Navigate the GUI with Shortcuts* to learn more about the shortcut icons.

Monitoring Graphs

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**.

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.

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)

Services			
Service	Description	Status	Actions
bsnmpd	SNMP Service	Running	  
dhcpcd	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. 28.6: Services Status

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 utilization of 9.96Mbit/s during a 1 minute period.

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 displayed 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.

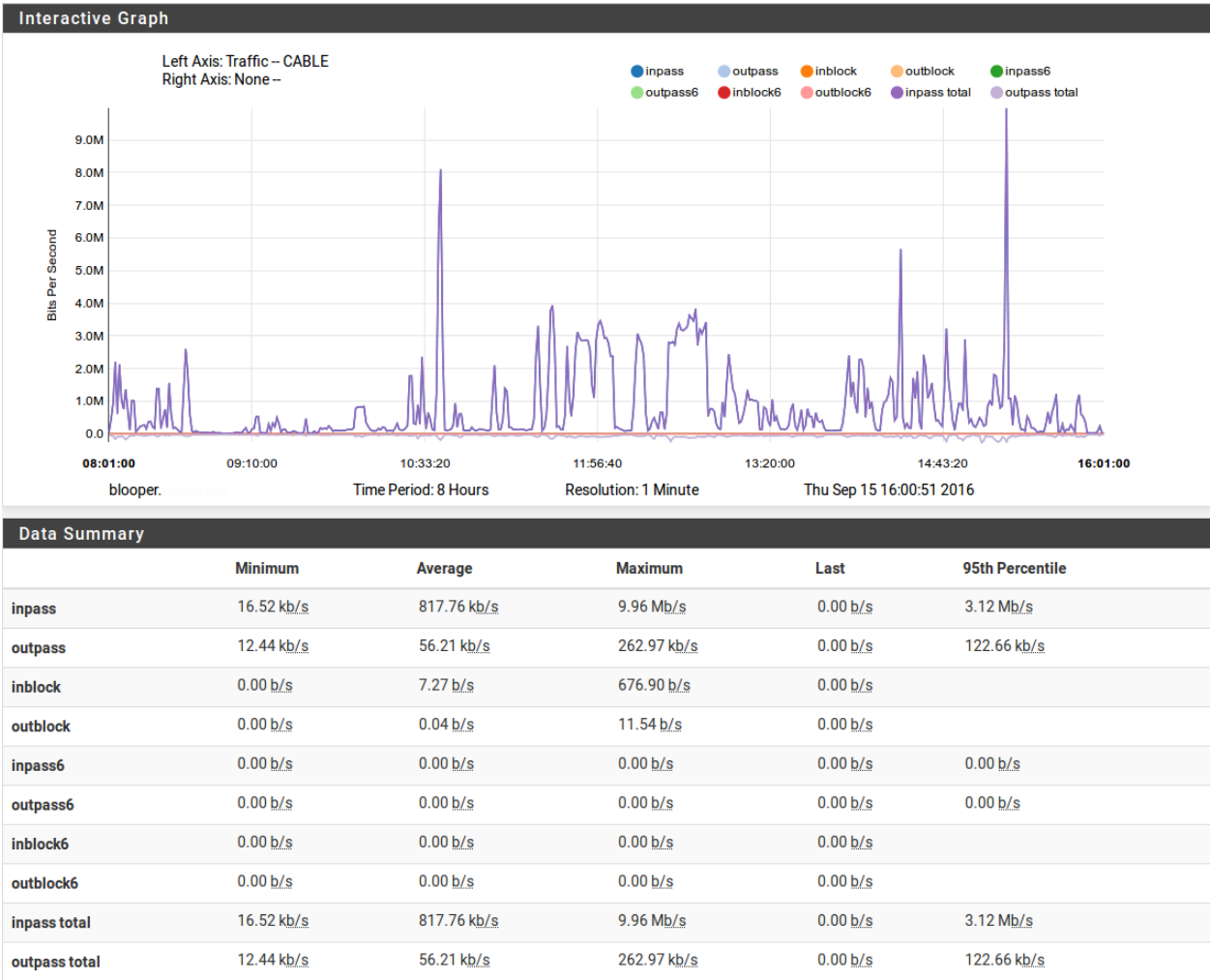


Fig. 28.7: WAN Traffic Graph

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 consolidated 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 possible 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 firewall.

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 throughput 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.

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. 28.8: Example 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.

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.

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.

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.

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.

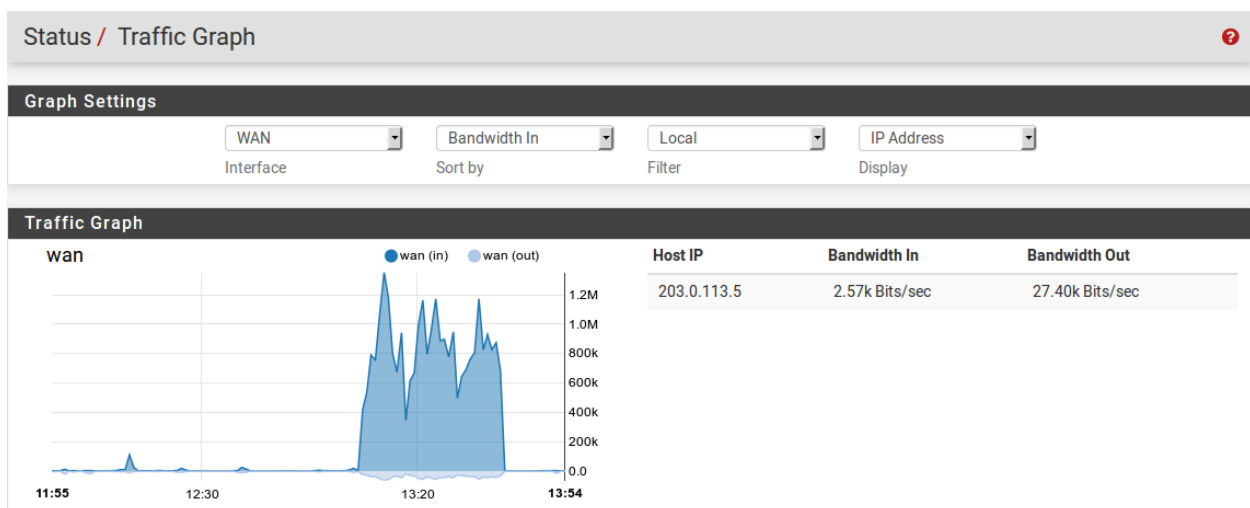


Fig. 28.9: Example LAN Graph

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](#).

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.

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.

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 and 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.

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:     090630PB4400XXXXXXX
LU WWN Device Id:  5 000cca 597XXXXXX
Firmware Version:  PB40C64G
User Capacity:     500,107,862,016 bytes [500 GB]
Sector Size:       512 bytes logical/physical
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 majority 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 15/40)
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.
 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 1
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
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.

SMTP and Growl Notifications

The status of the system can be reported passively using SMTP or Growl notifications. These notifications allow clients to receive alerts about system events without being logged into the firewall. Configuration and use of these mechanisms is covered in *Notifications*.

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.

Testing DNS

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 results of the DNS query are displayed on the page, along with supporting information and options.

The addresses returned by the DNS query are printed in the **Results** panel, along with the record type.

Next to the  **Lookup** button is a new button labeled  **Add alias**, which does exactly that: It creates an alias under **Firewall > Aliases** containing the results of the query as entries in the alias.

Underneath the results is a table showing 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, and links to external tools for looking up information about who owns the host or IP address.

Testing a TCP Port

The **Diagnostics > Test Port** page performs a simple TCP port connection test to see 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 during this test, it will only attempt to open a connection and optionally display the data sent back from the server.

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 host to be tested. This is a required field and must be a valid port number, meaning an integer between 1 and 65,535.

Source Port If needed, a manually specified source port for the query. This is not required 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.

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 given a hostname. If the connection is forced to IPv4 or IPv6 and the hostname does not contain a result using that protocol, it will result in an error. For example if forced to IPv4 and given a hostname that only returns an IPv6 IP address (AAAA record), it will not work.

The data and information that pfSense 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.

PACKAGES

Installing Packages

Packages are managed at **System > Packages**. The listings there, exemplified by Figure *Package Listing*, show all of the information about a package: Its name, category, version and status, a package information link, and a short description. Keep the installed packages to the bare minimum required for a deployment for extra security. The package list is presented in alphabetical order.

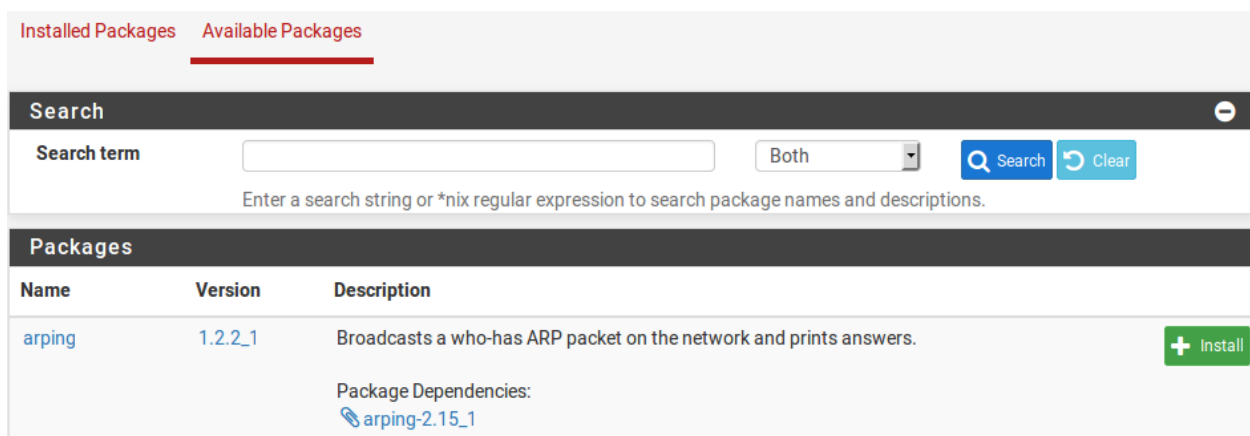


Fig. 29.1: Package Listing

Packages are installed as follows:

- Navigate to **System > Packages**
- Click the **Available Packages** tab
- Locate the package to install in the list
 - Optionally, search for a package by entering a value in the **Search term** box and clicking  **Search**
- Click the  **Install** button to the right of the package entry.
- Click  **Confirm** to proceed with the package installation

Once the installation is confirmed, the package installation screen is shown where the install progress is displayed (Figure *Post-Install Package Screen*).

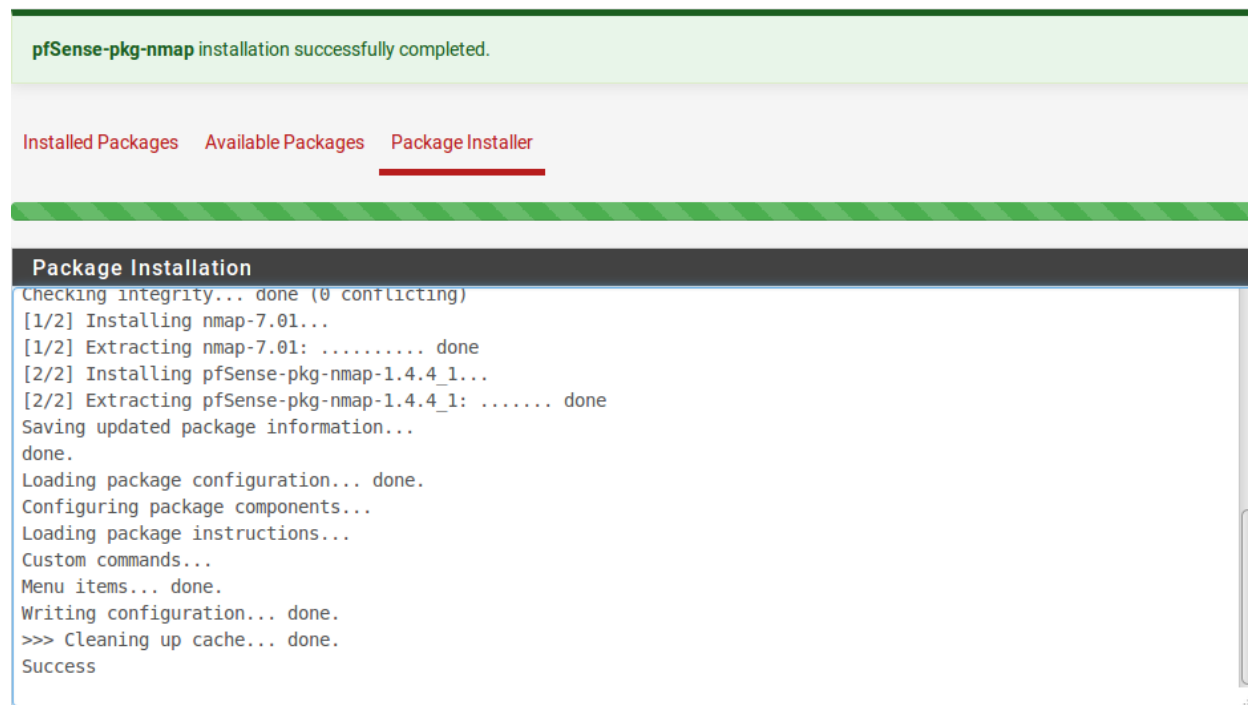


Fig. 29.2: Post-Install Package Screen

Reinstalling and Updating Packages

Packages are reinstalled and updated the same way they are installed:

- Navigate to **System > Packages**
- Click the **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 be highlighted stating the old and new versions.
- Click  to update or  to reinstall the package.
- Click  **Confirm** to proceed with the package reinstallation

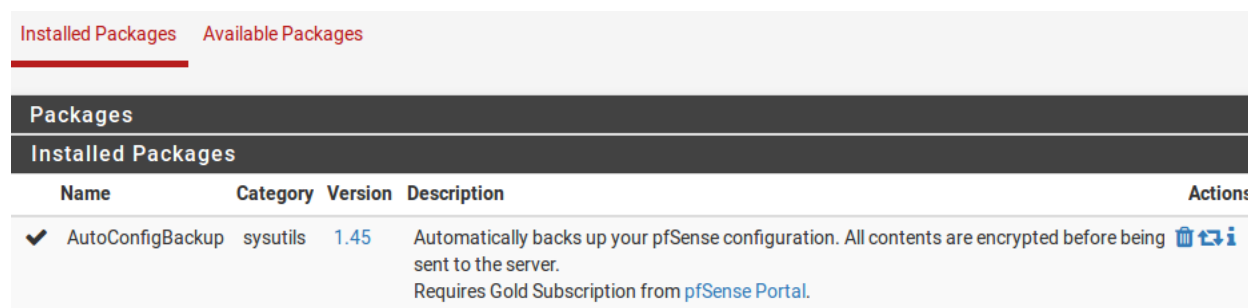


Fig. 29.3: Installed Package List

Uninstalling Packages

To uninstall a package:

- Navigate to **System > Packages**
- Click the **Installed Packages** tab
- Locate the package to uninstall in the list
- Click  to remove the package
- Click  **Confirm** to proceed with the package removal

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. For those interested in creating packages, resources are available on the pfSense Documentation Wiki, start with the [article about Developing Packages](#).

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.

A Brief Introduction to Web Proxies and Reporting: Squid, SquidGuard, and Lightsquid

This section is not meant to be a formal, detailed, and comprehensive how-to 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 has a simple single LAN and single WAN configuration. If problems occur during this configuration, visit the [pfSense Documentation Wiki](#) and [pfSense Forum](#) for additional guidance and assistance.

See also:

Customers with a [pfSense Gold Subscription](#) may access the [Hangouts Archive](#) 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.

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 If pfSense is running on a full install (NOT embedded/NanoBSD) and web access reporting is desired, 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 NanoBSD, enter 0 here.

Hard Disk Cache System If running NanoBSD, 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.

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 can work on full installs and NanoBSD, but blacklists may only be used on full installations. SquidGuard is capable of much more than will be covered in this section. Visit the [pfSense Documentation Wiki](#) and [pfSense 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](#).

Warning: Blacklists do not work properly on NanoBSD. Do not attempt to use them on that platform.

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**

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**.

Warning: This feature is not compatible with embedded/NanoBSD installs without manual modifications that are beyond what most users are capable of performing.

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.

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 book, but there is information on many of those tactics on the [pfSense Documentation Wiki](#) and [pfSense Forum](#).

The pfSense package system provides the ability to extend pfSense without adding bloat and potential security vulnerabilities to the base distribution. Packages are supported on full installs and a reduced set of packages are available on NanoBSD-based embedded installs.

Note: NanoBSD installs have the capability of running some packages, but due to the nature of the platform and its disk writing restrictions, some packages will not work and thus are not available for installation on that platform.

To see the packages available for the current firewall platform being utilized, browse to **System > Packages**, on the **Available Packages** tab.

Introduction to Packages

Many of the packages have been written by the pfSense community and not by the pfSense development team. 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 itself, like the OpenVPN Client Export Utility package which automatically creates VPN configuration files.

By far the most popular package available for pfSense is the Squid Proxy Server. It is installed more than twice as often as the next most popular package: Squidguard, which is a content filter that works with Squid to control access to web resources by users. Not surprisingly, the third most popular package is Lightsquid, which is a Squid log analysis package that makes reports of the web sites which have been visited by users behind the proxy.

Some other examples of available packages (which are not Squid related) are:

- Bandwidth monitors that show traffic by IP address such as ntopng, and Darkstat.
- Extra services such as FreeRADIUS.
- Proxies for other services such as SIP and FTP, and reverse proxies for HTTP or HTTPS such as HAProxy.
- 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, many others.
- Some items that were formerly in the base system but were moved to packages, such as RIP (routed)

As of this writing there are more than 40 different packages available; too many to cover them all in this book! The full list of packages that can be installed on a particular system is available from within any pfSense system by browsing to **System > Packages**.

The packages screen may take a little longer to load than other pages in the web interface. This is because the firewall fetches the package information from the pfSense package servers before the page is rendered to provide the most up-to-date package information. If the firewall does not have a functional Internet connection including DNS

resolution, this will fail and trigger a notification. If the package information has been retrieved previously, it will be displayed from cache, but the information will be outdated. This is usually caused by a missing or incorrect DNS server configuration. For static IP connections, verify working DNS servers are entered on the **System > General Setup** page. For those with dynamically assigned connections, ensure the DNS servers assigned by the ISP are functioning. This traffic will only go via the default gateway on the firewall, so ensure that gateway is up or change another active WAN gateway to be the default.

THIRD PARTY SOFTWARE AND PFSense

RADIUS Authentication with Windows Server

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.

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.

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

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**

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.

- 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**.

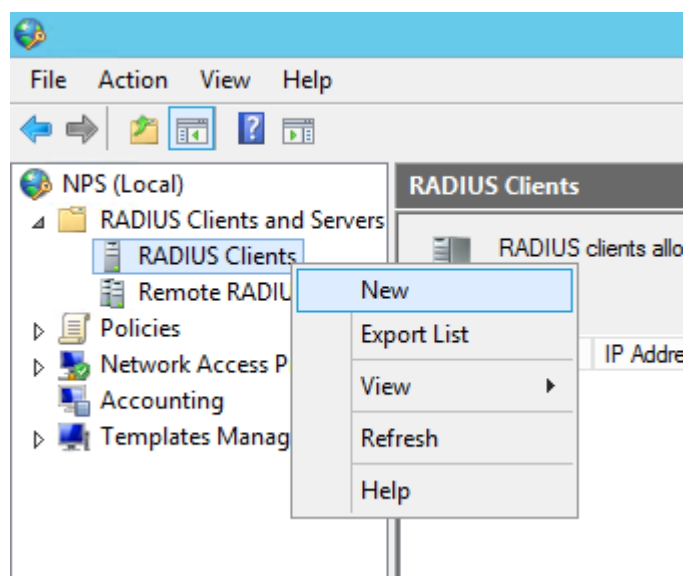


Fig. 30.1: Add New RADIUS Client

A screenshot of the 'New RADIUS Client' dialog box. The 'Settings' tab is active. It contains a checkbox 'Enable this RADIUS client' which is checked. Below it is a checkbox 'Select an existing template:' which is unchecked, followed by a dropdown menu. The 'Name and Address' section has a 'Friendly name:' field containing 'fw0' and an 'Address (IP or DNS):' field containing '198.51.100.2'. A 'Verify...' button is next to the address field. The 'Advanced' tab is also visible.

Fig. 30.2: Add New RADIUS Client Address

- Click OK.

Fig. 30.3: Add New RADIUS Client Shared Secret

The NPS configuration is now complete. The RADIUS Client is visible as in Figure *Listing of the RADIUS Client*.

Friendly Name	IP Address	Device Manufacturer	NAP-Capable	Status
 fw0	198.51.100.2	RADIUS Standard	No	Enabled

Fig. 30.4: Listing of the RADIUS Client

Refer to earlier sections in this book 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.

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.

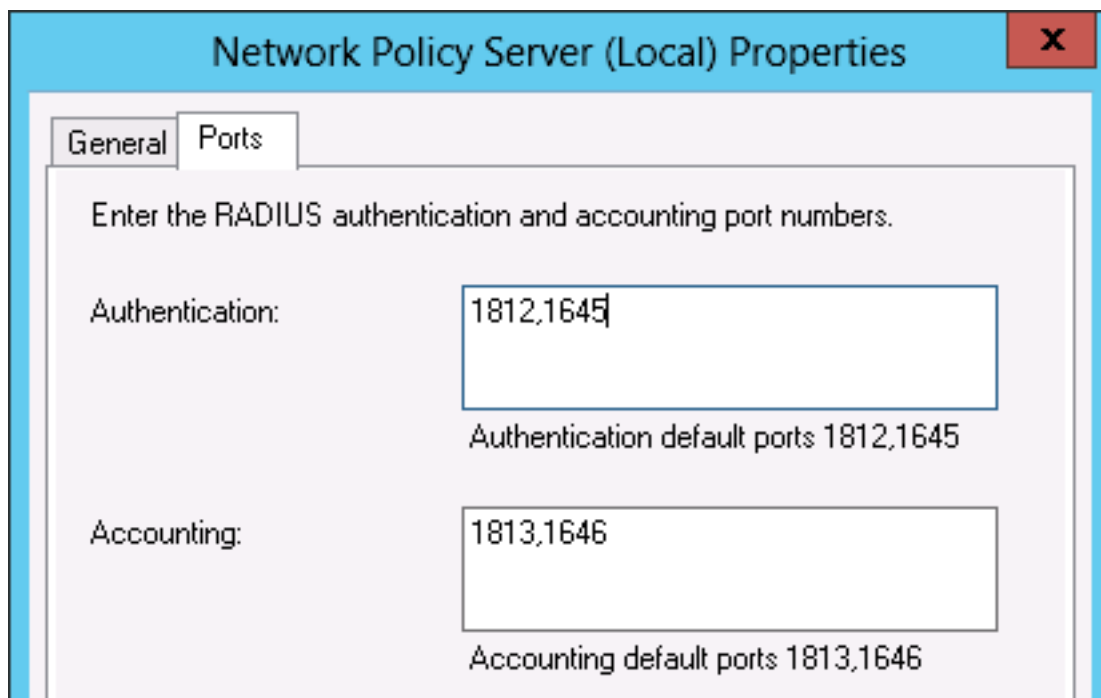


Fig. 30.5: NPS Ports

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.

Syslog Server on Windows with Kiwi Syslog

pfSense can send logs to an external server via the syslog protocol (*Remote Logging with Syslog*). For Windows users, [Kiwi Syslog Server](#) is a nice free option for collecting logs from pfSense firewalls. It can be installed as a service for long term log collection, or run as an application for shorter term needs. It is compatible with both server and desktop versions of Windows 2000 and newer. The installation is straight forward, and doesn’t require much configuration. Help can be found in its documentation after installation.

Using Software from FreeBSD’s Ports System (Packages)

Because pfSense is based on FreeBSD, many familiar FreeBSD packages are available for use by veteran FreeBSD system administrators.

Warning: Installing software this way is not for the inexperienced, as it could have unintended side-effects, and is not recommended nor supported.

Many parts of FreeBSD are not included, so library and other issues can occur when attempting to use software installed in this manner. pfSense 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’s pre- built package repository.

Concerns/Warnings

Several important concerns must be considered by any administrator before deciding to install additional software to a pfSense firewall, especially software that is not a sanctioned package.

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 pfSense packages are not immune to this problem either. Any additional service is another potential attack vector.

Performance Concerns

Most hardware running pfSense can handle the traffic load with which they are tasked. If the firewall hardware has horsepower to spare, it may not hurt the system 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 before trying to install anything.

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 will need to 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 can also install additional web pages that are not protected by the authentication process on pfSense. Test any installed software to ensure that access is protected or filtered in some manner.

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 pfSense 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 files such as these.

Installing Packages

To install a package, the proper package site must be used. pfSense is compiled against a specific FreeBSD RELEASE branch, and has only a specific set of packages hosted on the project servers.

Packages located in the pfSense package repository, including some FreeBSD software packages that are not a part of pfSense, can be installed using `pkg install` directly:

```
# pkg install iftop
```

Or use a full URL to a `pkg add` to add them from the FreeBSD package servers:

```
# pkg add http://pkg.freebsd.org/freebsd:10:x86:64/quarterly/All/iftop-1.0.p4.txz
```

The package will be downloaded and installed, along with any needed 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 can introduce problems with package dependencies, especially if a package depends on another piece of software that already exists on pfSense 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 pfSense firewall. Due to the complexity of this topic, it won't be covered here.

Maintaining Packages

The following command prints a list of all currently installed packages, including pfSense packages and parts of the pfSense base system:

```
# pkg info
```

To delete an installed package, pass its full name or use a wildcard:

```
# pkg_delete iftop-1.0.p4
# pkg_delete pstree-\*
```

Configure 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.

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; };
};
```

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
```

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.

While this book is focused on pfSense, there are a number of third party software packages that can be configured to interoperate with pfSense or augment its functionality. In this context, *third party software* refers to software available from other vendors or sources which can be used together with pfSense, but is not considered part of the “pfSense system”. These are different from pfSense packages, which are extra software that runs on the pfSense system and integrates into the system’s GUI.

PACKET CAPTURING

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 > (assign)** 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 31.1: Real Interfaces vs. Friendly Names

Real/Physical Name	Friendly Name
<code>enc0</code>	IPsec, encrypted traffic
<code>ovpnc0 ... ovpnc<x>, ovpsn0 ... ovpsn<x></code>	OpenVPN, encrypted traffic (Clients, Servers)
<code>pppoe0 ... pppoe<x>, poes0 ... poes<x></code>	PPPoE WAN, PPPoE Server
<code>l2tp0 ... l2tp<x>, l2tps0 ... l2tps<x></code>	L2TP WAN, L2TP Server
<code>lo0</code>	Loopback Interface
<code>pfsync0</code>	pfsync interface – used internally
<code>pflog0</code>	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.

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.

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 [Using tcpdump from 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.

Using tcpdump from 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 with pfSense, 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 book. Indeed, entire books have been written on this subject alone. For those with a thirst for more than basic knowledge in this area, some recommendations for additional reading are provided at the end of this chapter. This section is intended to provide an introduction to this topic, and leave the reader with enough knowledge for basic troubleshooting.

tcpdump command line 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 31.2: 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 31.3: Example Uses of tcpdump -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
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, auth type none, intv
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: 10.0.64.2
23:30:05.915110 00:0c:29:0b:c3:ed > 00:13:d4:f7:73:d2, ethertype IPv4 (0x0800), length 170: 10.0.64.2
23:30:05.915396 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800), length 60: 10.0.64.1
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
23:30:06.065200 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800), length 60: 10.0.64.1
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 igbl tcp port 80
```

Capture all DNS traffic Capture all DNS traffic (usually UDP, but some queries use TCP):

```
# tcpdump -ni igbl 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 `igbl` interface:

```
# tcpdump -ni igbl 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 `igbl` 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 igbl -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 igbl 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 auth:
14:50:08.436849 IP 198.51.100.12 > 224.0.0.18: CARPv2-advertise 36: vhid=11 advbase=1 advskew=0 auth:
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
```

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
```

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]
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,
15:52:46.151814 (authentic,confidential): SPI 0xc0afb14d: IP 10.7.0.1 > 10.3.0.1: ICMP echo reply, id
```

```
15:52:47.154243 (authentic,confidential): SPI 0xcd77e085: IP 10.3.0.1 > 10.7.0.1: ICMP echo request, id=154243, seq=154243
15:52:47.154843 (authentic,confidential): SPI 0xc0afb14d: IP 10.7.0.1 > 10.3.0.1: ICMP echo reply, id=154243, seq=154243
```

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
```

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.

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

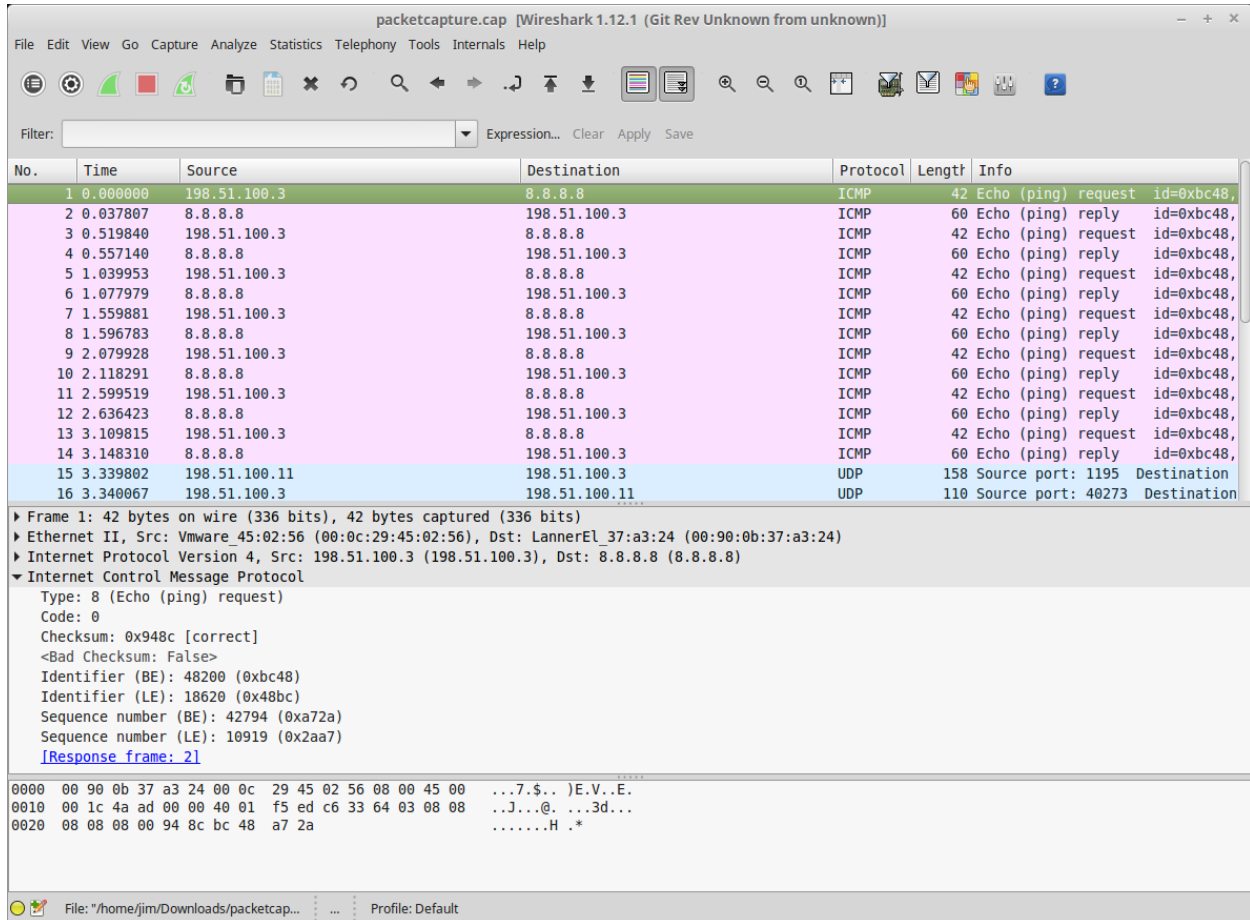


Fig. 31.1: Wireshark Capture View

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. 31.2: 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 2.3.1-RELEASE-p5 (amd64 full-install) on clara ***
[...]
```

After confirming the SSH connection works, start the remote capture as follows:

```
# wireshark -k -i <(ssh root@192.168.1.1 tcpdump -i igbl -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.

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 book.

[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 WebGUI, with `tcpdump` at the command line in a shell, and using Wireshark.

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.

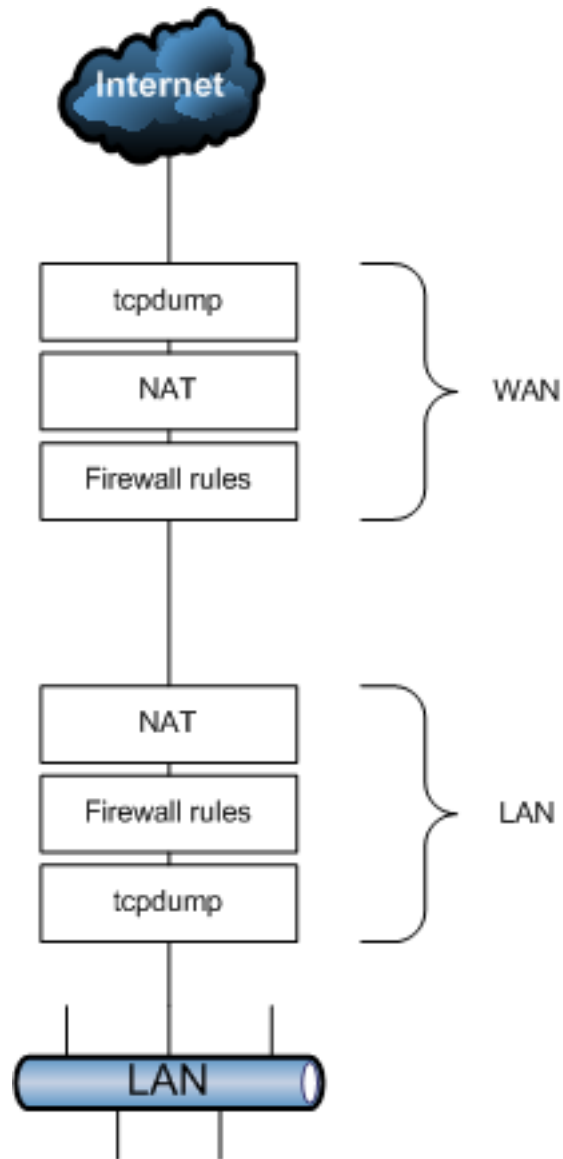


Fig. 31.3: Capture Reference

MENU GUIDE

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. (e.g. update from pfSense 2.3.2 to 2.4). 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**.

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)

(assign) 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](#).

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](#).

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 [DHCP & DHCPv6 Relay](#).

DHCP server Configures the DHCP service which provides automatic IP address configuration for clients. See [IPv4 DHCP Server](#).

DHCPv6 Relay Configures the DHCP relay service for IPv6 which proxies DHCPv6 requests from one network segment to another. See [DHCP & DHCPv6 Relay](#).

DHCPv6 Server & RA Configures the DHCP service for IPv6 and Router Advertisements which provide automatic IPv6 address configuration for clients. See [IPv6 DHCP Server and Router Advertisements](#).

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 (“dyn dns”) 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](#).

Load Balancer Configures the Load Balancer, which balances *incoming* connections across multiple local servers. See [Server Load Balancing](#).

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](#).

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](#).

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](#).

- Load Balancer** Shows the status of the server Load Balancer pools. See [Viewing load balancer status](#).
- 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](#).

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](#).
- 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 [Testing DNS](#).
- 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 [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 [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 [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. This guide will help to quickly identify the purpose of a given menu option, and refer to places in the book 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.