# pfSense: The Definitive Guide Version 2.1

## The Definitive Guide to the pfSense Open Source Firewall and Router Distribution

**Christopher M. Buechler**
**Jim Pingle**

# pfSense: The Definitive Guide Version 2.1: The Definitive Guide to the pfSense Open Source Firewall and Router Distribution

by Christopher M. Buechler and Jim Pingle

Based on pfSense Version 2.1

Publication date 2013


**Abstract**

The official guide to the pfSense open source firewall distribution.

# Table of Contents

# List of Figures

# List of Tables

# 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 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 — and most of that is running cables and explaining the difference between "inside" and "outside." pfSense's extensive documentation and user community offers me an easy answer to questions — "did you look that up?" If pfSense doesn't support a feature, chances are I couldn't support it either. But pfSense supports everything I could ask for, and with a friendly interface to boot. The wide userbase means that features are tested in many different environments and generally "just work," 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, PPTP, 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

# Preface

Welcome to *The Definitive Guide to pfSense*. Written by pfSense co-founder Chris Buechler and pfSense consultant Jim Pingle, this book covers installation and basic configuration through advanced networking and firewalling with the popular open source firewall and router distribution.

This book is designed to be a friendly step-by-step guide to common networking and security tasks, plus a thorough reference of pfSense's capabilities. *The Definitive Guide to pfSense* covers the following subjects:

- An introduction to pfSense and its features.
- Hardware and system planning.
- Installing and upgrading pfSense.
- Using the web-based configuration interface.
- Backup and restoration.
- Firewalling fundamentals and defining and troubleshooting rules.
- Port forwarding and Network Address Translation (NAT).
- General networking and routing configuration.
- Bridging, Virtual LANs (VLANs), and Multi-WAN.
- Virtual Private Networks using IPsec, PPTP, and OpenVPN.
- Traffic shaping and load balancing.
- Wireless networking and captive portal setups.
- Redundant firewalls and High Availability.
- Various network related services.
- System monitoring, logging, traffic analysis, sniffing, packet capturing, and troubleshooting.
- Software package and third-party software installations and upgrades.

At the end of this book, you'll find a menu guide with the standard menu choices available in pfSense and a detailed index.

# Authors

## Chris Buechler

Chris is one of the founders of the pfSense project, and one of its most active developers. He has been working 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 from small organizations to Fortune 500 companies and large public sector organizations. He currently makes a living helping organizations with pfSense related needs including network design, deployment planning, configuration assistance, conversion from existing firewalls, development and more. He is based in Louisville, Kentucky USA and provides services for customers around the world. 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 over ten years, professionally for the past eight years. Now a full-time employee of BSD Perimeter, LLC, he provides global support for pfSense commercial support subscribers. As a system administrator with HPC Internet Services, a local ISP in Bedford, Indiana, USA he works with FreeBSD servers, various routing equipment and circuits, and of course pfSense-based firewalls both internally and for many customers. Jim has a Bachelor's degree in Information Systems from Indiana-Purdue Fort Wayne, and graduated in 2002. He also contributes to several Open Source projects besides pfSense, most notably RoundCube Webmail and glTail.

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

# Acknowledgements

This book, and pfSense 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, with 14 people contributing routinely and considerably enough to obtain commit access. Hundreds have contributed financially, with hardware, and other needed resources. Thousands more have done their part to support the project by helping others on the mailing list, forum, and IRC. Our thanks to everyone who has done their part to make the project the great success it has become.

## Book Cover Design

Thanks to Holger Bauer for the design of the cover. Holger was one of the first contributors to the project, having done much of the work on theming, graphics, and is the creator of the backgrounds we have used on our presentations at six BSD conferences over the past five years.

## pfSense Developers

The current active pfSense development team, listed in order of seniority.

- Co-Founder Scott Ullrich
- Co-Founder Chris Buechler
- Bill Marquette
- Holger Bauer
- Erik Kristensen
- Seth Mos
- Scott Dale
- Martin Fuchs
- Ermal Luçi
- Matthew Grooms
- Mark Crane
- Rob Zelaya
- Renato Botelho
- Erik Fonnesbeck

- Warren Baker
- Luiz Costa

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

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

# 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 lead to countless long days and 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 the second as well. 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 your feedback for this book and the pfSense distribution. Please send your suggestions, criticism and/or praise for The Definitive Guide to pfSense book to <book@pfsense.org>. The publisher's webpage for the book is at http://www.reedmedia.net/books/pfsense/.

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 a few conventions are used 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 | Apply Changes |
| Prompt for input | `Do you want to proceed?` |
| Input from the user | **`Rule Description`** |
| File Names | `/boot/loader.conf` |
| Names of commands or programs | **gzip** |
| Commands Typed at a shell prompt | `# ls -l` |
| Items that must be replaced with values specific to your setup | *`192.168.1.1`* |
| Special Notes |  **Note**<br>Watch out for this! |

Long literal lines in output examples may be split with the # (hookleftarrow). Long shell command-line examples may be split using the backslash (\) for shell line continuation.

# Chapter 1. Introduction

pfSense is a free, open source customized distribution of FreeBSD tailored for use as a firewall and router, entirely managed in 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, and, in fact, the majority of the user base has never used FreeBSD outside of pfSense. In addition to being a powerful, flexible firewalling and routing platform, it includes a long list of related features and a package system allowing 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 proven in countless installations ranging from small home networks protecting a single computer to large corporations, universities and other organizations protecting thousands of network devices.

## Project Inception

This project was founded in 2004 by Chris Buechler and Scott Ullrich. Chris had been contributing to m0n0wall for some time before that, and found it to be a great solution. However, while thrilled with the project, many users longed for more capabilities than can be accommodated in a project strictly focused towards embedded devices and their limited hardware resources. Enter pfSense. In 2004, there were numerous embedded solutions with 64 MB RAM that couldn't be accommodated with the desired feature set of pfSense, so pfSense expanded to also work on more capable PC and server type hardware. However, modern embedded hardware is also well supported and popular with pfSense today.

## What does pfSense stand for/mean?

The project ran for a couple months with no name. In fact, the FreeBSD jail that used to run our CVS server was called `projectx` up until we migrated away from CVS to git several years ago.

Scott and Chris were the only two members of the project at the time, as its founders. We ran through numerous possibilities, with the primary difficulty being finding something with domain names available. Scott came up with pfSense, pf being the packet filtering software used, as in making sense of PF. Chris' response was less than enthusiastic. But after a couple weeks with no better options, we went with it. It was even said "well, we can always change it."

Since then, a name change was considered amongst the developers, without gaining any traction as most people were indifferent and nobody felt a compelling need for change. In mid 2007, a discussion of naming was initiated by a blog post, and the overwhelming response from the community via email and blog comments was "keep the name!"

## Why FreeBSD?

Since many of the core components in pfSense come from OpenBSD, you may wonder why we chose FreeBSD rather than OpenBSD. There were numerous factors under consideration when choosing an OS for this project. This section outlines the primary reasons for choosing FreeBSD.

### Wireless Support

We knew wireless support would be a critical feature for many users. At the time this project was founded in 2004, OpenBSD's wireless support was very limited. Its driver support was much more limited than FreeBSD's, and it had no support for important things such as WPA (Wi-Fi Protected Access) and WPA2 with no plans of ever implementing such support at the time. Some of this has changed since 2004, but FreeBSD remains ahead in wireless capabilities.

Due in large part to the work being done by Adrian Chadd on FreeBSD's wireless support, it remains ahead of other BSD-based distributions even today.

## Network Performance

FreeBSD's network performance is significantly better than that of OpenBSD. For small to mid sized deployments, this generally isn't of any concern, as upper scalability is the primary issue in OpenBSD. One of the pfSense developers manages several hundred OpenBSD PF firewalls, and has had to switch his high load systems over to FreeBSD PF systems to handle the high packets per second rate required in portions of his network. This has become less of an issue in OpenBSD since 2004, but still holds true.

FreeBSD also has a multi-processor version of PF in newer development versions, that may be used in future versions of pfSense, allowing for greater scalability. NetBSD has a multi-processor PF work-alike in progress, NPF, but as of this writing it was still fairly young in its development and features.

## Familiarity and ease of fork

Since the pfSense code base started from m0n0wall, which is based on FreeBSD, it was easier to stay with FreeBSD. Changing the OS would require modifying nearly every part of the system. Scott and Chris, the founders, are also most familiar with FreeBSD and had previously worked together on a now-defunct commercial FreeBSD-based firewall solution. This in and of itself wasn't a compelling reason, but combined with the previous two factors it was just another thing to point us in this direction.

## Alternative Operating System Support

At this time, there are no plans to support any other operating systems, simply for reasons of resource constraints. It would be a considerable undertaking to port to any of the other BSDs as we do rely on some functionality that is only available in FreeBSD, which would have to be completely refactored.

# Common Deployments

pfSense is used in about every type and size of network environment imaginable, and is almost certainly suitable for your network whether it contains one computer, or thousands. This section will outline the most common deployments.

## Perimeter Firewall

The most common deployment of pfSense is as a perimeter firewall, with an Internet connection plugged into the WAN side, and the internal network on the LAN side.

pfSense accommodates networks with more complex needs, such as multiple Internet connections, multiple LAN networks, multiple DMZ networks, etc.

Some users also add BGP (Border Gateway Protocol) capabilities to provide connection redundancy and load balancing. This is described further in Chapter 12, *Routing*.

## LAN or WAN Router

The second most common deployment of pfSense is as a LAN or WAN router. This is a separate role from the perimeter firewall in midsized to large networks, and can be integrated into the perimeter firewall in smaller environments.

### LAN Router

In larger networks utilizing multiple internal network segments, pfSense is a proven solution to connect these internal segments. This is most commonly deployed via the use of VLANs with 802.1Q trunking, which will be described in Chapter 14, *Virtual LANs (VLANs)*. Multiple Ethernet interfaces are also used in some environments.

### Note

In environments requiring more than 3 Gbps of sustained throughput, or more than 500,000 packets per second, no router based on commodity hardware offers adequate performance. Such environments need to deploy layer 3 switches (routing done in hardware by the switch) or high end ASIC-based routers. As commodity hardware increases in performance, and general purpose operating systems like FreeBSD improve packet processing capabilities in line with what new hardware capabilities can support, scalability will continue to improve with time.

## WAN Router

For WAN services providing an Ethernet port to the customer, pfSense is a great solution for private WAN routers. It offers all the functionality most networks require and at a much lower price point than big name commercial offerings.

# Wireless Access Point

Many deploy pfSense strictly as a wireless access point. Wireless capabilities can also be added to any of the other types of deployments.

# Special Purpose Appliances

Many deploy pfSense as a special purpose appliance. The following are four scenarios we know of, and there are sure to be many similar cases of which we are not aware. Most any of the functionality of pfSense can be utilized in an appliance-type deployment. You may find something unique to your environment where this type of deployment is a great fit. As the project has matured, there has been considerable focus on using it as an appliance building framework, especially in the 2.0 release. Some pre-packaged special purpose appliances will be made available in the future, but you can use pfSense to make your own appliances even without pre-packaged versions.

## VPN Appliance

Some users drop in pfSense as a VPN appliance behind an existing firewall, to add VPN capabilities without creating any disruption in the existing firewall infrastructure. Most pfSense VPN deployments also act as a perimeter firewall, but this is a better fit in some circumstances.

## DNS Server Appliance

pfSense offers a DNS (Domain Name System) server package based on TinyDNS, a small, fast, secure DNS server. It isn't laden with features, so it isn't able to be used for some purposes such as Microsoft Active Directory, but it's a great fit for hosting public Internet DNS. Remember the DNS vulnerability chatter in July 2008? Daniel J. Bernstein, the author of TinyDNS, is credited with the original idea and implementation of randomized source ports in the DNS resolver, the resolution to that vulnerability. In fact, TinyDNS was the only major DNS server that did not need to be patched in July 2008. It has used randomized source ports since its inception. Several years ago, Bernstein even put $1000 USD of his own money on the line for the first person to find a privilege escalation security hole. It remains unclaimed. If you're hosting only public Internet DNS, TinyDNS should be strongly considered. The pfSense package also adds failover capabilities, and easier support for IPv6 addresses (AAAA and PTR) in the GUI.

## Sniffer Appliance

One user was looking for a sniffer appliance to deploy to a number of branch office locations. Commercial sniffer appliances are available with numerous bells and whistles, but at a very significant cost especially when multiplied by a number of branch locations. pfSense offers a web interface for **tcpdump** that allows the downloading of the resulting pcap file when the capture is finished. This

enables this company to capture packets on a branch network, download the resulting capture file, and open it in Wireshark [http://www.wireshark.org] for analysis.

pfSense is not nearly as fancy as commercial sniffer appliances, but offers adequate functionality for many purposes at a vastly lower cost.

For more information on using the packet capture features of pfSense, see Chapter 30, *Packet Capturing*.

In pfSense 2.0 you can also add a span port as part of a bridge, which retransmits a copy of every frame received on the bridge. This can be used to passively monitor the traffic on another device, such as a system running snort for traffic analysis, without disrupting traffic flow.

### DHCP Server Appliance

One user deploys pfSense installs strictly as DHCP (Dynamic Host Configuration Protocol) servers to hand out IP addresses for its network. In most environments this probably does not make much sense, due to the limitations of the pfSense GUI for advanced configuration of the ISC DHCP daemon. It is much improved in 2.0 and 2.1, but still incapable of performing many complex, but possible, DHCP server tasks. In this user's case, the staff was already familiar and comfortable with pfSense and this enabled further deployments without additional training for the administrators, which was an important consideration in this deployment.

# Versions

This section describes the different pfSense releases available currently and historically.

## 2.1 Release

The pfSense 2.1 [http://doc.pfsense.org/index.php/2.1_New_Features_and_Changes] release is the current recommended release for all installations. The improvements in 2.1 are mainly focused around adding IPv6 support, which has been quite a major undertaking, affecting nearly every part of the system. pfSense 2.1 is based on FreeBSD 8.3-RELEASE. Because this is the latest official release, it is the only release that will receive bug fixes and security updates. You can find the current recommended release by browsing to www.pfsense.org/versions [http://www.pfsense.org/versions]. Unless otherwise stated, features mentioned in this book may only be available on 2.1, but the fundamentals should be similar across most recent versions.

## 2.0.3 Release

pfSense 2.0.3 [http://doc.pfsense.org/index.php/2.0.3_New_Features_and_Changes] was released to fix issues discovered in 2.0.2. Primarily, it fixed issues with obtaining DNS servers from PPP type WANs, and issues some were seeing with captive portal and traffic graphs. It is still based on FreeBSD 8.1-RELEASE.

## 2.0.2 Release

pfSense 2.0.2 [http://doc.pfsense.org/index.php/2.0.2_New_Features_and_Changes] was released as a security and bug fix/enhancement release. It addressed several Security Advisories from FreeBSD, some XSS/CSRF issues, and brought in some other bug fixes and enhancements for things that were found after 2.0.1 was released. It is still based on FreeBSD 8.1-RELEASE.

## 2.0.1 Release

pfSense 2.0.1 [http://doc.pfsense.org/index.php/2.0.1_New_Features_and_Changes] was released as a security release and minor bug fix/enhancement. It addressed a security issue with the certificates

generation and brought in some bug fixes and enhancements for things that were found after 2.0 was released. It is still based on FreeBSD 8.1-RELEASE.

## 2.0 Release

The pfSense 2.0 [http://doc.pfsense.org/index.php/2.0_New_Features_and_Changes] release (formerly known as 1.3) contained numerous significant enhancements over previous versions that are covered throughout this book. It is based on FreeBSD 8.1-RELEASE. The release cycle for 2.0 ran much longer than expected, but it is a very ambitious release in terms of added features and improvements. As such, it required much more extensive testing and debugging. It is our hope to shorten the release cycle on future versions, perhaps releasing every 6 months or so.

## 1.2.3 Release

Still quite popular, this previous release is used in many areas that have not yet upgraded to 2.0 for various reasons. The 1.2.3 release provided a number of bug fixes and enhancements from 1.2.2, and updated the base OS to FreeBSD 7.2. References in this book to 1.2 mostly include every 1.2.x release, though some things mentioned in this book only exist in 1.2.3 and later releases.

## 1.2, 1.2.1, 1.2.2 Releases

1.2 was the first stable release in the 1.2 line of releases, and was made available on February 25, 2008. The 1.2.1 update provided a number of bug fixes and some minor security fixes, and updated the base OS to FreeBSD 7.0. The 1.2.2 release added a few bug fixes.

## 1.0 Release

This was the first release of pfSense classified as stable. It was released on October 4, 2006, with a follow up 1.0.1 bug fix release on October 20, 2006. Though we know of installs still running some early alpha versions and countless sites still running 1.0, it is no longer supported and we strongly recommend all users upgrade to 2.1. 1.0.1 contains several minor security vulnerabilities fixed in either 1.2 or 1.2.1.

## Snapshot Releases

The pfSense snapshot server builds a new image from the code currently in our source code repository either every 24 hours, or sooner if there is a commit to the tree. These are primarily for developers and users testing bug fixes at the request of a developer. Snapshots may not always be available, depending on the point in the release cycle. Shortly after the 1.2 release, the snapshots were taken offline as the build infrastructure was updated to FreeBSD 7.0 and the 1.3 (at the time, later renumbered to 2.0) release was prepared for the first publicly available releases. The same was done between 2.0 and 2.1. Similar situations may exist in the future. You can see what snapshots, if any, are available by visiting the snapshot server [http://snapshots.pfsense.org]. The snapshot builds are commonly stopped if there is a large batch of commits ongoing, or other build debugging is being performed.

# Platforms

pfSense offers three platforms suitable for three different types of deployments. This section covers each of the available platforms, and the environments in which they are best suited for use.

## Live CD (including the USB memstick image)

The Live CD platform allows you to run directly from the CD (or USB memory stick) without installing to a hard drive or Compact Flash card. References to the "Live CD" throughout the book also refer to the USB memstick image. The configuration can be saved on a floppy disk or USB flash drive. The CD is not frequently accessed after boot since the system runs primarily from RAM at that

point, but should not be removed from a running system. In most circumstances, this should only be used as an evaluation of the software with your particular hardware. Many people do use it long term, but we recommend using full installs instead. Live CD users cannot use packages, and the historical performance graphs are lost at restart.

# Full Install

The Live CD includes an installer option to install pfSense to the hard drive on your system. This is the preferred means of running pfSense. The entire hard drive must be overwritten; dual booting with another OS is not supported. Full installs are recommended for most deployments. From download statistics we can surmise at least 80% of all pfSense deployments are full installs. Most of the developers use full installs primarily if not entirely. Hence it's the most widely tested and best supported version. It does not have some of the limitations of the other platforms.

# Embedded

The embedded version is specifically tailored for use with any hardware using Compact Flash (CF) rather than a hard drive. CF cards can only handle a limited number of writes, so the embedded version runs read only from CF, with read/write filesystems as RAM disks. Even with that limitation, they are widely supported in embedded hardware and via IDE-to-CF converters. Though CF cards are smaller than a traditional ATA hard drive connector, their pin arrangement is similar and they are electrically compatible. This makes it easier to implement for devices which already support IDE. CF being solid state media, you also don't have to worry about the potential failure of a spinning disk.

Embedded systems are popular for many reasons, but the most compelling ones are that they typically have few if any moving parts, and they consume much less power and produce less heat than larger systems while still performing well enough for the needs of most networks. In this case, less moving parts means less points of failure, less heat, and they can run completely silent.

Historically, embedded has been a second class citizen with pfSense, as full installs have been the primary focus of the project. This has changed with the current generation of embedded, based on NanoBSD.

One drawback of embedded systems is that some of the historical graphing data in RRDtool is lost if the system is not shut down cleanly. For example, a power outage will cause some graph data loss. This does not affect functionality, but will leave blank spots in your historical graphs. You can setup periodic backups of the RRD data, which mitigates this risk somewhat. RRD data can also be backed up with the config.

## Old Embedded (pre-1.2.3 release)

Packages were not supported on the older embedded versions 1.2.2 and earlier. Older embedded upgrades also did not always work reliably. The only 100% guaranteed reliable means of upgrading embedded installs was to backup the configuration, re-flash the CF, and restore the configuration. These limitations have all been eliminated in the new embedded setup.

## NanoBSD Embedded

NanoBSD is a standard means of building FreeBSD in an embedded friendly fashion. It supports dual firmware, and is reliably upgradeable. At the time of this writing, NanoBSD embedded is fully functional and being used in production. The 1.2.3 release began using this embedded methodology, at which time the old embedded was discontinued. 2.0 and more recent releases only use the new embedded methodology.

In addition to multiple firmware support enabling switching between two different installs, this brings two additional important benefits. Packages are supported, for those suitable for an embedded environment. It also allows cross-building for hardware architectures other than x86, with MIPS and potentially ARM platforms being supported in the future.

# Interface Naming Terminology

This section describes the interface naming terminology used in pfSense and FreeBSD. Most people are familiar with the two basic network divisions: "WAN" and "LAN", but there can be as many segments as you can imagine. You are only limited by the number of interfaces (or VLANs) you have at your disposal. Starting with pfSense 2.0, you can rename every interface to whatever name you want. In previous versions, you could only rename optional interfaces, not WAN/LAN.

While discussing the interface names, the topic of network segmentation also comes to mind. It is a good practice to keep different sets of systems apart from each other. For example, you don't want your publicly-accessible web server on the same network as your LAN. If the server was compromised, the attacker could easily reach any system on your LAN. If you have dedicated database servers, these can be isolated from everything else and secured from everything except the servers which need database access. As with the previous example, a compromised web server would not endanger the database servers nearly as much as if they were on the same segment without a firewall in between them.

## LAN

The LAN interface is the first internal interface on your firewall. Short for Local Area Network, it is most commonly the private side of a router which often utilizes a private IP address scheme. In small deployments, this is typically the only internal interface.

## WAN

The WAN interface is used for your Internet connection, or primary Internet connection in a multi-WAN deployment. Short for Wide Area Network, it is the untrusted public network outside of your router. Connections from the Internet will come in through the WAN interface.

## OPT

OPT or Optional interfaces refer to any interfaces connected to local networks other than LAN. OPT interfaces are commonly used for second LAN segments, DMZ segments, wireless networks and more.

## OPT WAN

OPT WAN refers to Internet connections using an OPT interface, either those configured for DHCP or specifying a gateway IP address. This is discussed in detail in Chapter 15, *Multiple WAN Connections*.

## DMZ

Short for demilitarized zone. The term was borrowed from its military meaning, which refers to a sort of buffer between a protected area and a war zone. In networking, it is an area where your public servers reside that is reachable from the Internet via the WAN, but is also isolated from the LAN so that a compromise in the DMZ does not endanger systems in other segments.

Some companies misuse the term "DMZ" in their firewall products in reference to 1:1 NAT on the WAN IP which exposes a host on the LAN. There is more information on that subject in the section called “1:1 NAT on the WAN IP, aka "DMZ" on Linksys”.

# FreeBSD interface naming

FreeBSD names its interfaces by the network driver used, followed by a number starting at 0 and incrementing by one for each additional interface using that driver. For example, a common driver is `em`, used by Intel Pro/1000 cards. The first Pro/1000 card in a system will be `em0`, the second is `em1`, etc. Other common ones are `igb` (Also Intel Pro/1000), `bge` (various Broadcom chipsets), `rl` (Realtek

8129/8139), amongst numerous others. If your system mixes a Pro/100 card and a Realtek 8139, your interfaces will be `fxp0` and `rl0` respectively. Interface assignments and naming are further covered in Chapter 4, *Installing and Upgrading*.

# Finding Information and Getting Help

This section offers guidance on finding information in this book, and on pfSense in general, as well as providing resources on where to get further help if needed.

## Finding Information

The easiest way to find information on a specific topic in this book is to check the Index. All the most common features and deployments of pfSense are covered in this book, and the Index will help you find the section or sections where a specific topic is covered.

If you cannot find the information you seek in this book, there is a wealth of additional information and user experiences available on the various pfsense.org sites. The best way to search all these sites is to head to Google, type in the terms you are looking for, and append **`site:pfsense.org`** to your query. This will search the website, forum, our Redmine site, wikis, etc. — all official sources of information. There is a wealth of information available on the forum, and this is the best way of searching it. This will also locate information in the freely available portions of this book.

## Getting Help

In pfSense 2.0 and later, there is a help icon on almost every page, . Clicking this help icon will take you to an associated page on our documentation wiki with additional information about the page you are viewing, or the general function you are using. The pfSense project offers several other ways to get help, including a forum [http://forum.pfsense.org], documentation wiki [http://doc.pfsense.org], mailing lists and IRC (Internet Relay Chat, ##pfSense on irc.freenode.net). Commercial support is also available via subscription from the founders of the pfSense project on the pfSense Portal [https://portal.pfsense.org]. You can find more information on all these support avenues on the Obtaining Support [http://www.pfsense.org/support] page on the pfSense site. Many of these are also linked from the pfSense 2.0 GUI under the Help menu.

# Chapter 2. Networking Concepts

While this is not an introductory networking book, there are certain networking concepts that are important to understand. This portion of the book will not provide adequate coverage for those lacking basic fundamental networking knowledge. If you do not possess this knowledge, you will likely need to seek additional introductory networking material.

Readers with significant knowledge of public and private IP addressing, IP subnetting, CIDR notation and CIDR summarization can skip to the next chapter. There will be some mention of IPv6 related topics here, but more in-depth coverage may be found in the section called “IPv6”. We will typically refer to traditional IP addresses as IPv4 addresses for clarity. Most functions will work with either IPv4 or IPv6 addresses, except where noted otherwise, so if we refer to an IP address in general, it can mean one of either address family.

## Brief introduction to OSI Model Layers

Throughout this book, we will refer to different pieces of information as pertaining to a specific network layer, such as "Layer 2" or "Layer 3". These layers are defined by the OSI model, which is covered in many networking texts and, of course, on Wikipedia (http://en.wikipedia.org/wiki/OSI_model). Without delving too deep into what can exist on each layer, we will go over each layer very briefly. Some concepts maybe a bit oversimplified here, so it is best to seek out more detailed sources if you need further information on the subject.

| | |
|---|---|
| Layer 1 — Physical | Most often, this means cabling, such as Cat 5/6 cable, Fiber, Coax, etc. |
| Layer 2 — Data Link | Typically means Ethernet or another similar protocol that is being spoken on the wire. In this book, we often refer to layer 2 as meaning your Ethernet switches, or related topics such as ARP and MAC addresses that function at layer 2. |
| Layer 3 — Network Layer | Here you have 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, Bittorrent, etc. |

And last but not least, sometimes people in the technology world also jokingly refer to Layer 8, meaning the user.

For more tangible information about the layers and how they interact visually, you may be interested in the Network Protocols Map poster by Javvin (http://www.javvin.com/map.html) which illustrates how a connection from any of several different protocols can flow through each layer of the OSI model.

# Understanding Public and Private IP Addresses

There are two main types of IP addresses found in most networks — public and private.

## Private IP Addresses

Private IP addresses are those within a reserved subnet, for internal use only. The network standard RFC 1918 [http://www.faqs.org/rfcs/rfc1918.html] defines reserved IPv4 subnets for use in private networks (Table 2.1, “RFC 1918 Private IP Address Space”), and RFC 4193 [http://tools.ietf.org/html/rfc4193] defines Unique Local Addresses (ULA) for IPv6 (Table 2.2, “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, which are then connected to the Internet through a firewall or router implementing Network Address Translation (NAT), such as pfSense. For IPv6, you generally use Global Unicast Addresses (GUA), which are fully routed, even on your internal networks, without NAT. NAT will be explained further in Chapter 11, *Network Address Translation*.

### Table 2.1. RFC 1918 Private IP Address Space

| CIDR Range | IP Address Range |
|---|---|
| 10.0.0.0/8 | 10.0.0.0 - 10.255.255.255 |
| 172.16.0.0/12 | 172.16.0.0 - 172.31.255.255 |
| 192.168.0.0/16 | 192.168.0.0 - 192.168.255.255 |

### Table 2.2. RFC 4193 Unique Local Address Space

| Prefix | IP Address Range |
|---|---|
| fc00::/7 | fc00:: - fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff |

For IPv4 there have historically been other reserved ranges such as 1.0.0.0/8 and 2.0.0.0/8 but these are not permanently reserved like the RFC 1918 addresses and many have since been allocated as the IPv4 pool has been exhausted. It was tempting to use these, and some networks may still use them, but because those networks have been allocated, continuing to use those IPs will result in the inability to reach any systems in those new networks. You should also avoid using 169.254.0.0/16, which according to RFC 3927 is reserved for "Link-Local" autoconfiguration — but *should not* be assigned by DHCP or manually. There is more than enough address space set aside by RFC 1918, as shown in Table 2.1, “RFC 1918 Private IP Address Space”, so there is little incentive to deviate from that list. We have encountered networks with all manner of improper addressing, and it will lead to problems — it isn't a question of "if", but "when" problems will occur. If you find yourself working on an existing network using an improper address space, it is best to correct the addressing as soon as possible. A complete list of special-use IPv4 networks may be found in RFC 3330 [http://tools.ietf.org/html/rfc3330].

## Public IP Addresses

Public IP addresses are those assigned by your ISP for all but the biggest networks. Networks requiring hundreds or thousands of public IP addresses commonly have address space assigned directly from the Regional Internet Registry covering their region of the world. Regional Internet Registries are the organizations that oversee allocation and registration of public IP address in their designated region of the world.

Most residential Internet connections come with a single public IPv4 address, while most business class connections come with an option of using multiple public IPs if necessary. A single public IP is adequate in many circumstances and can be used in conjunction with NAT to connect hundreds of

privately addressed systems to the Internet. Content throughout this book will help you determine the number of public IPs your network requires.

Most IPv6 deployments will give the end user at least a /64 prefix network to use as a routed internal network, which is roughly $2^{64}$ IPv6 addresses, or 18 *quintillion* addresses for your site, fully routed from the Internet with no need for NAT.

# IP Subnetting Concepts

When configuring the TCP/IP settings on a device, a subnet mask (Or prefix length for IPv6) must be specified. This mask enables the system to determine which IP addresses are on the local network, and which must be reached by a gateway in the system's routing table. The default LAN IP 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 the section called “Understanding CIDR Subnet Mask Notation”.

# IP Address, Subnet and Gateway Configuration

The TCP/IP configuration of a host consists of three primary things — address, subnet mask (or prefix length for IPv6) and gateway. The IP address and subnet mask combined is how the host knows which IP addresses are on its local network. For any addresses outside the local network, traffic is sent (routed) to the configured default gateway which must know how to reach the desired destination. An exception to this rule is a static route, which instructs a router or system on how to contact specific non-local subnets reachable via locally connected routers. This list of gateways and static routes is kept on each host in its routing table. To see the routing table used by pfSense, see the section called “Viewing Routes”. More information about routing can be found in Chapter 12, *Routing*.

In a typical pfSense deployment, hosts will be assigned an IP address within the LAN range of pfSense, the same subnet mask as the LAN interface of pfSense, and use pfSense's LAN IP as their default gateway. The same applies to hosts connected to an interface other than LAN, using the appropriate configuration for the interface to which the device is connected.

Hosts within a single network communicate directly with each other with no involvement from the default gateway. This means no firewall, including pfSense, can control host to host communication within a network segment. If this functionality is required, hosts either need to be segmented via the use of multiple switches or VLANs, or equivalent switch functionality like PVLAN needs to be employed. VLANs are covered in Chapter 14, *Virtual LANs (VLANs)*.

# Understanding CIDR Subnet Mask Notation

pfSense uses a subnet mask format with which you may not be familiar. Rather than the common 255.x.x.x, it uses CIDR (Classless InterDomain Routing) notation.

You can refer to Table 2.3, “CIDR Subnet Table” to find the CIDR equivalent of your subnet mask.

### Table 2.3. 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 | 0 | 1/128th |
| 255.255.255.252 | /30 | 4 | 2 | 1/64th |
| 255.255.255.248 | /29 | 8 | 6 | 1/32nd |

| Subnet Mask | CIDR Prefix | Total IP Addresses | Usable IP Addresses | Number of /24 networks |
|---|---|---|---|---|
| 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 |

## So where do these CIDR numbers come from anyway?

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 a /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 the CIDR subnet table indicates how many addresses

a given CIDR mask will summarize. 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, static routes, and more.

IPs or networks that can be contained within a single CIDR mask are known as CIDR summarizable.

When designing a network you should ensure all private IP subnets in use at a particular location are CIDR summarizable. For example, if you need three /24 subnets at one location, use a /22 network subnetted into four /24 networks. The following table shows the four /24 subnets you can use with the subnet 10.70.64.0/22.

### Table 2.4. 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 helps keep routing more manageable for multi-site networks (those connected to another physical location via the use of a private WAN circuit or VPN). With CIDR summarizable subnets, you have one route destination that covers all the networks at each location. Without it, you have several different destination networks per location.

Now, if you aren't a subnetting guru, you're probably wondering how the heck I came up with the previous table. Start by choosing a CIDR prefix for your network, according to the number of networks you will require. Then pick a /24 network that you want to use. For that example, I chose 10.70.64.0/24. I know from memory that x.x.64.0/24 will be first /24 network in a /22, but you don't have to pick the first network. You can easily calculate this using the tools available on the subnetmask.info [http://www.subnetmask.info] website.

One of the tools will convert from dotted decimal to CIDR mask, and vice versa, this function is shown in Figure 2.1, “Subnet Mask Converter”. If you didn't have Table 2.3, “CIDR Subnet Table” from earlier in this chapter in front of you, you could convert your chosen CIDR prefix to dotted decimal notation using this tool. Enter a CIDR prefix and click the Calculate button to its right, or enter a dotted decimal mask and click the Calculate button to its right.

### Figure 2.1. Subnet Mask Converter


Armed with the dotted decimal mask, now go to the Network/Node Calculator section. Put in the subnet mask and one of the /24 networks you want to use. Then click Calculate. The bottom boxes will fill in, and show you the range covered by that particular /24, which you can see in Figure 2.2, “Network/Node Calculator”. In this case, the network address will be 10.70.64.0/22, and you can see that the usable /24 networks will be 64 through 67. "Broadcast address" isn't relevant terminology when you are using this tool to determine a CIDR range, that is simply the highest address within the range.

**Figure 2.2. Network/Node Calculator**


## Finding a matching CIDR network

IP Ranges in the format of x.x.x.x-y.y.y.y are supported in Aliases in 2.0. An IP range is automatically converted to the equivalent set of CIDR blocks. See the section called “Aliases” for more information.

If you don't necessarily need an exact match, you can plug in numbers to the Network/Node Calculator to get close to your desired summarization.

# Broadcast Domains

A broadcast domain is the portion of a network sharing the same layer two network segment. In a network with a single switch, 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, implied in the previous sentence, is that you can run both IPv4 and IPv6 networks within a single broadcast domain. This is called dual stack, and is a common and useful technique to have 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 will give the same effect, such as a DHCP relay which relays DHCP requests into another interface's broadcast domain. More information on broadcast domains and how to combine them can be found in Chapter 13, *Bridging*.

# IPv6

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 had allocated all of it's space to regional internet registries [http://www.nro.net/news/ipv4-free-pool-depleted] (RIRs). In turn, these RIR allocations have run out in some locations such as APNIC (Asia/Pacific) and RIPE (Europe) for /8 networks and though some smaller allocations are still available, it is increasingly difficult to obtain new IPv4 address space in these regions.. ARIN (North America) is due to run out sometime in 2014 [http://www.potaroo.net/tools/ipv4/index.html].

IPv6 was created as a replacement for IPv4 to account for this. It has been available in some forms since the 1990's, but due to factors like inertia, complexity, and the cost of developing or purchasing compatible routers and software, its uptake has been slow until the last few years [http://www.google.com/ipv6/statistics.html#tab=ipv6-adoption] when the IPv4 allocations started drying

up, and even then it's been rather slow [http://arstechnica.com/business/2013/01/ipv6-takes-one-step-forward-ipv4-two-steps-back-in-2012/].

Over the years, support for IPv6 in software, operating systems, and routers has improved so the situation is primed to get better, but it's still up to ISPs to start delivering IPv6 connectivity to users. It's left the world in a bit of a catch-22: Content providers are slow to provide IPv6 because few users have it. At the same time, users don't have it because there isn't a lot of IPv6 content, and even less that is only available over IPv6. User's don't know they need it, so they don't demand the service from their ISPs, and everyone gets stuck.

Some providers are experimenting with Carrier Grade NAT (CGN) to stretch their IPv4 networks farther [http://www.techweekeurope.co.uk/news/plusnet-ipv4-ipv6-nap-networking-104349], essentially sticking 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, though the applications typically found on mobile devices don't tend to care that it's in place, or there isn't an extra layer since to them it seems as if they're just behind a typical SOHO router style NAT. More brokenness is observed when such connectivity is used as a firewall's WAN, or when tethering on a PC, or in some cases just attempting to use a traditional IPsec VPN without NAT-T, or PPTP. These technologies are only delaying the inevitable, but may at least help some ISPs transition users over to IPv6 where needed. If at all possible, avoid using a provider that employs CGN.

This section is not meant to be an exhaustive resource on IPv6 information, but a primer to get you started. 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, and it's worth using as a starting point if you need more information on IPv6. There are also many good books on IPv6 available, but be careful when purchasing a book to find one with a recent revision, as 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.

# Basics

Essentially, IPv6 allows for exponentially more IP space than IPv4 did. IPv4 uses a 32-bit address, which allows for $2^{32}$ or just over 4 billion addresses, less if you remove the sizable reserved blocks and IPs burned by subnetting. IPv6 uses a 128-bit address, which is $2^{128}$ or $3.403 \times 10^{38}$ IPs. The standard size IPv6 subnet defined by the IETF is a /64, which contains $2^{64}$ IPs, or 18.4 *quintillion* addresses. You can fit the entire IPv4 space inside of a typical IPv6 subnet many times over with room to spare.

Aside from the obvious addressing concerns that IPv6 solves, there are some other improvements in the specification that can go overlooked. One of the more subtle ones is that there are no IPs lost to subnetting. With IPv4, you lose two IP addresses per subnet to account for a null route and broadcast IP address. This is not the case in IPv6, as broadcast is handled via the same mechanisms used for multicast, which involve special addresses that are not inside the subnet but sent to the entire network segment. Some other improvements include IPsec being part of the spec, larger potential packet sizes, and other design elements that make it easier for IPv6 to be handled in routers than IPv4 at the packet level.

Other than those factors, one of the biggest differences between IPv4 and IPv6 is that in IPv6, everything is routed. There should be no NAT involved at all. So each end user device IP address can, unless stopped by a firewall, be directly reached by any other. This can be very difficult for people to grasp who are used to having their LAN exist with a specific private subnet and then doing NAT to whatever the external address happens to be.

As we'll cover in the remainder of this section, there are some fundamental differences in how IPv6 operates in comparison to IPv4, but mostly these are just that: differences. Some things are simpler than IPv4, others are slightly more complicated, but mostly it's simply different and may take some adjustment to get used to the new style. The differences are at layer 2 (ARP vs. NDP for instance)

and layer 3 (IPv4 vs. IPv6 addressing) only. 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

Some people mistakenly consider NAT as an additional firewall protection mechanism, in place of traditional firewall rules and controls. When it comes to IPv6, this NAT isn't there, or in some cases like with Network Prefix Translation (NPt) the end user IP is essentially visible anyhow, so it becomes even more important to consider proper firewall controls. You will no longer need to setup port forwards to open up access to devices on your LAN, just firewall rules.

A consequence of everything being routable is that you need to ensure things take the proper and expected path when reaching your network. For example, you need to make sure that LAN to LAN traffic takes your encrypted VPN connections and does not route directly to the remote site.

See the section called “IPv6 VPN and Firewall Rules” for a more in-depth discussion on IPv6 firewall concerns with respect to VPN traffic.

# Requirements

In order to use IPv6, you must have some way to reach an IPv6-enabled network. If you are lucky and your ISP delivers you native IPv6 connectivity directly, that is ideal. Some ISPs are deploying a dual stack configuration, meaning they deliver IPv4 and IPv6 alongside each other on the same transport. If you are not so lucky, then there are other ways you can obtain access. Some ISPs use tunneling or deployment types that can get IPv6 to you indirectly, or you can use a third party tunnel broker such as Hurricane Electric's tunnelbroker service [http://www.tunnelbroker.net] or the similar service from SixXS [http://www.sixxs.net].

In addition to the service, you must also have software that supports using IPv6. pfSense 2.1 is the first IPv6-capable version published, so upgrading to that version is a must. Beyond that, you must also ensure your client operating systems and applications support using IPv6. Many common operating systems and applications support it without problems, but when in doubt, check and see if any updates are needed. Microsoft Windows has supported IPv6 since at least Windows XP (though newer versions handle it much better), OS X has used it for some time, FreeBSD and Linux both support it in the operating system as well. Most web browsers and mail clients support it, and many other common applications, so long as they are on recent versions.

Some mobile operating systems have varying levels of support for IPv6. Android and iOS both support IPv6, but they only support stateless autoconfiguration for obtaining an IP address and not DHCPv6. IPv6 is part of the LTE spec, so any mobile device new enough to support LTE networks should support IPv6 as well.

# IPv6 WAN Types

Most of the details can be found in the section called “IPv6 WAN Types”, but some of the most common ways of deploying IPv6 are:

| | |
|---|---|
| Static Addressing | Native and either using IPv6 on its own or in a dual stack configuration alongside IPv4. |
| DHCPv6 | Address automatically obtained by DHCPv6 to an upstream server. Optionally, prefix delegation may also be used with DHCPv6 to deliver a routed subnet to a DHCPv6 client. |
| Stateless address autoconfiguration (SLAAC) | Automatically determines the IPv6 address by consulting router advertisement messages and then programmatically 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 to use. It may be useful for appliance modes. |
| 6RD Tunnel | A method of tunneling IPv6 traffic inside IPv4, used by ISPs for rapid IPv6 deployment. |
| 6to4 Tunnel | Similar to 6RD, but with different mechanisms and limitations. |
| GIF Tunnel | Not technically a "wan" type directly, but very common to use. Customer builds an IPv4 GIF tunnel to a provider, and then tunnels IPv6 traffic over that. |

# 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`. The 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, such as `0001:0001:0001:0001:0001:0001:0001:0001` could be written as `1:1:1:1:1:1:1:1`.

Any number of zeroes may be compressed by using `::` but this can only be done once in an IP address to avoid ambiguity. A good example of this is localhost, which is `0000:0000:0000:0000:0000:0000:0000:0001`, and compresses to `::1`. Any time you see `::` in an IPv6 address, you can know that the values between are all zeroes. So if you have an IP address such as `fe80:1111:2222:0000:0000:0000:7777:8888`, it 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, so 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 this would be ambiguous as 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 a lot more freedom to design how you want to address your devices.

On servers that I intend to use multiple IP aliases on for things like virtualhosts, jails, etc, I use the seventh section of the IPv6 address to denote the server, and then use the 8th section for individual IPv6 aliases. This groups all of the IPs for a single host together in an easy to recognize way. 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 people also like to have a little fun with their IPv6 addresses, using the hexadecimal letters and number/letter equivalents to made words out of their IP addresses. You can find lists of hexadecimal words around the web [http://nedbatchelder.com/text/hexwords.html], and then use them to form IP addresses that may be a bit more memorable, such as `2001:db8:1:1::dead:beef`.

## Decimal vs. Hexadecimal Confusion

Owing to the fact that IPv6 addresses are hexadecimal based, there can be some confusion when making consecutive IP addresses. Now it may not really be important to use consecutive IP addresses efficiently on IPv6, but if you are looking to keep things tight/clean in a list, it's worth noting. Since hexadecimal values are base 16, instead of base 10 like decimal, the IP 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. Is this really

important? That's debatable and entirely up to you and the design of the network. For some, it's easier to avoid using the hexadecimal digits where they can if it makes things simpler for them.

The main place this comes up is when designing a dual stack network, and you want to keep one section of the IPv6 address the same as its IPv4 counterpart. Given that all of the IPv4 address space can be expressed in IPv6, feel free to craft addresses however you see fit.

# IPv6 Subnetting

IPv6 subnetting can actually be easier than IPv4, it's just different. Want to divide or combine a subnet? Just add or chop off digits. No need to calculate where inside a range of IPs a subnet starts/ends, usable IPs, null route/broadcast, etc.

Where IPv4 had a CIDR mask to denote the subnet mask, IPv6 calls it a Prefix Length, and is often shortened to just Prefix. This number denotes how many bits of the address define the network in which the address exists. Most commonly, the prefixes you'll find in use on IPv6 are multiples of four, as seen in Table 2.5, “IPv6 Subnet Table”, but they can be any number between 0 and 128.

Using multiples of four for the prefix length makes it much easier to make IPv6 subnets. Using that convention, all you need to do is add or remove digits from the IPv6 address and adjust the prefix by a multiple of 4 and you made a bigger/smaller subnet. For reference, see Table 2.5, “IPv6 Subnet Table” which shows a table of possible IPv6 addresses and how they look, and how many IPs can be found inside each size subnet. Some of the values gets quite large, so it's difficult to type them all out in a way that will fit on a page!

### Table 2.5. 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 |

| Prefix | Subnet Example | Total IP Addresses | # of /64 nets |
|---|---|---|---|
| 92 | xxxx:xxxx:xxxx:xxxx:xxxx:xxx:: | $2^{36}$ (68,719,476,736) | 0 |
| 96 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:: | $2^{32}$ (4,294,967,296) | 0 |
| 100 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x:: | $2^{28}$ (268,435,456) | 0 |
| 104 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xx:: | $2^{24}$ (16,777,216) | 0 |
| 108 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxx:: | $2^{20}$ (1,048,576) | 0 |
| 112 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:: | $2^{16}$ (65,536) | 0 |
| 116 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:x:: | $2^{12}$ (4,096) | 0 |
| 120 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xx:: | $2^{8}$ (256) | 0 |
| 124 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxx:: | $2^{4}$ (16) | 0 |
| 128 | xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx | $2^{0}$ (1) | 0 |

A /64 is a standard size IPv6 subnet as defined by the IETF, and it the smallest subnet you can use locally if you desire to use autoconfiguration.

Typically your ISP will assign you a /64 or smaller between you and them, just to connect your networks together, and then they will route you an additional network for use on your LAN. How large of an allocation you receive depends upon your ISP, but it's not uncommon to see end users receive at least a /64, and in some cases up to a /48.

If you do not have native IPv6 connectivity but you use a tunnel service such as Hurricane Electric's tunnelbroker.net, they will allocate you a /48 to use as you see fit, in addition to a routed /64 subnet and a /64 interconnect.

Typically you'd use the first /64 in the large subnet as your LAN, but it can be divided any way you like: A /64 for LAN, a /64 for a VPN tunnel subnet, a /64 for a DMZ, a /64 for a guest network, etc.

# Special IPv6 Subnets

As with IPv4, there are some special reserved networks that you might see from time to time, or that have special uses (or preventions from use). A full list of these can be found in the Wikipedia IPv6 article [http://en.wikipedia.org/wiki/IPv6_address#Special_addresses], but a few important ones are shown in Table 2.6, “IPv6 Special Networks and Addresses”.

### Table 2.6. IPv6 Special Networks and Addresses

| Network | Purpose |
|---|---|
| 2001:db8::/32 | Documentation prefix, used for examples, as you might 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.

NDP works by sending special ICMPv6 packets to special multicast addresses and handles tasks for not just neighbor discovery, but also for router solicitations, and the handling of route redirects similar to IPv4's ICMP redirects.

pfSense 2.1 automatically adds firewall rules on IPv6 enabled interfaces to allow NDP to work properly. You can view all current known neighbors on IPv6 by visiting Diagnostics → NDP Table in the firewall GUI.

# Router Advertisements

In IPv6, you locate a router 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, and respond to router solicitations sent by clients on the network. When acting as a client (WAN interfaces), pfSense will accept RA messages from upstream routers. When acting as a router, pfSense can provide RA messages to clients on its internal networks. See the section called “Router Advertisements (Or: "Where is the DHCPv6 gateway option#")” for more details.

# Address Allocation

You can allocate addresses to clients using static addressing, SLAAC (the section called “Router Advertisements (Or: "Where is the DHCPv6 gateway option#")”), DHCP6 (the section called “IPv6 DHCP Server and Router Advertisements”), or via other tunneling methods such as OpenVPN. Which method you choose depends on what your potential clients support and where they are located on your network.

## DHCP6 Prefix Delegation

DHCP6 Prefix Delegation delivers a routed IPv6 subnet to a DHCP6 client, for use on its internal side. A WAN type interface can be set to receive a prefix over DHCP6 (the section called “DHCP6”, the section called “Track Interface”), or if you are acting at the edge of a large network, you can provide prefix delegation to other routers inside your network (the section called “DHCPv6 Prefix Delegation”).

# IPv6 and NAT

NAT is frowned upon in IPv6, but there are some situations that call for its use, such as Multi-WAN for IPv6 on residential or SOHO/Small/Medium business networks.

Gone is the traditional type of port translated NAT (PAT) where many addresses are translated using ports on a single external IP. Some routers do offer NAT66 to handle that kind of transformation, but it is somewhat rare.

Straight network address translation, akin to 1:1 NAT, does exist and it is called Network Prefix Translation or NPt for short. This is available in pfSense under Firewall → NAT on the NPt tab. NPt will take one prefix and translate it to another. So you could move `2001:db8:1111:2222::/64` to be `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. For more on NPt, see the section called “IPv6 Network Prefix Translation (NPt)”.

Though tempting for users to ask about as a transition mechanism, NAT64 also ranges from being ill-advised to non-existent in router support. NAT64 is used to allow IPv6 hosts to communicate with an IPv4 host. We do not yet currently have support for NAT64, though it may come in the future. The opposite direction, NAT46, is not really viable and likely won't ever exist in a usable way.

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, so where possible it's best not to rely on them.

# IPv6 and pfSense

Even with as much work as we managed to do for pfSense 2.1, there are still some areas that lack support for IPv6. Typically this isn't something we could have avoided due to other limitations, but

also some things are either not used quite so often or are more complicated and held back for the next release.

Obviously the major functionality for a network router is there: Addressing, firewalling, DHCP, CARP/HA, IPsec and OpenVPN, and more. In general it is safe to assume, unless noted otherwise, that IPv6 is supported in a given area or feature.

We have a complete list of IPv6 features and what does or does not work with IPv6 available in a Google Docs spreadsheet [https://docs.google.com/spreadsheet/ccc?key=0AojFUXcbH0ROdHlKV2F5SENULWk2NTVvQTBtQ2M0dEE&usp=sharing].

Some noteworthy areas that do *not* yet support IPv6 as of pfSense 2.1 are: Captive Portal, most DynDNS providers, UPnP, and PPTP.

### Note

If your system was upgraded from a previous version of pfSense (1.2.x, 2.0.x), you will need to allow IPv6 traffic using the master switch at System → Advanced on the Networking tab. Check the Allow IPv6 option if it is not already checked, then Save.

## pfSense Packages

Support is packages is a bit less complete than the base system. Many packages are maintained by the community and so it will take longer for packages to catch up, assuming some software packages even support IPv6.

For example, some packages like nmap have no problem supporting IPv6, but Quagga for OSPF needs some GUI work to allow it to function. The Quagga software itself supports OSPFv3 for IPv6, but the pfSense package GUI does not yet have support for controlling that part of the software.

There is a pfSense 2.1 packages spreadsheet that contains [https://docs.google.com/spreadsheet/ccc?key=0AojFUXcbH0ROdEFZcC1GbEV6ekpqcXFEZTE5TGpzaXc] a listing of packages known to support or not support IPv6. When it comes to the package system it's best to assume a package does not support IPv6 unless noted otherwise.

# Connecting with a Tunnel Broker Service

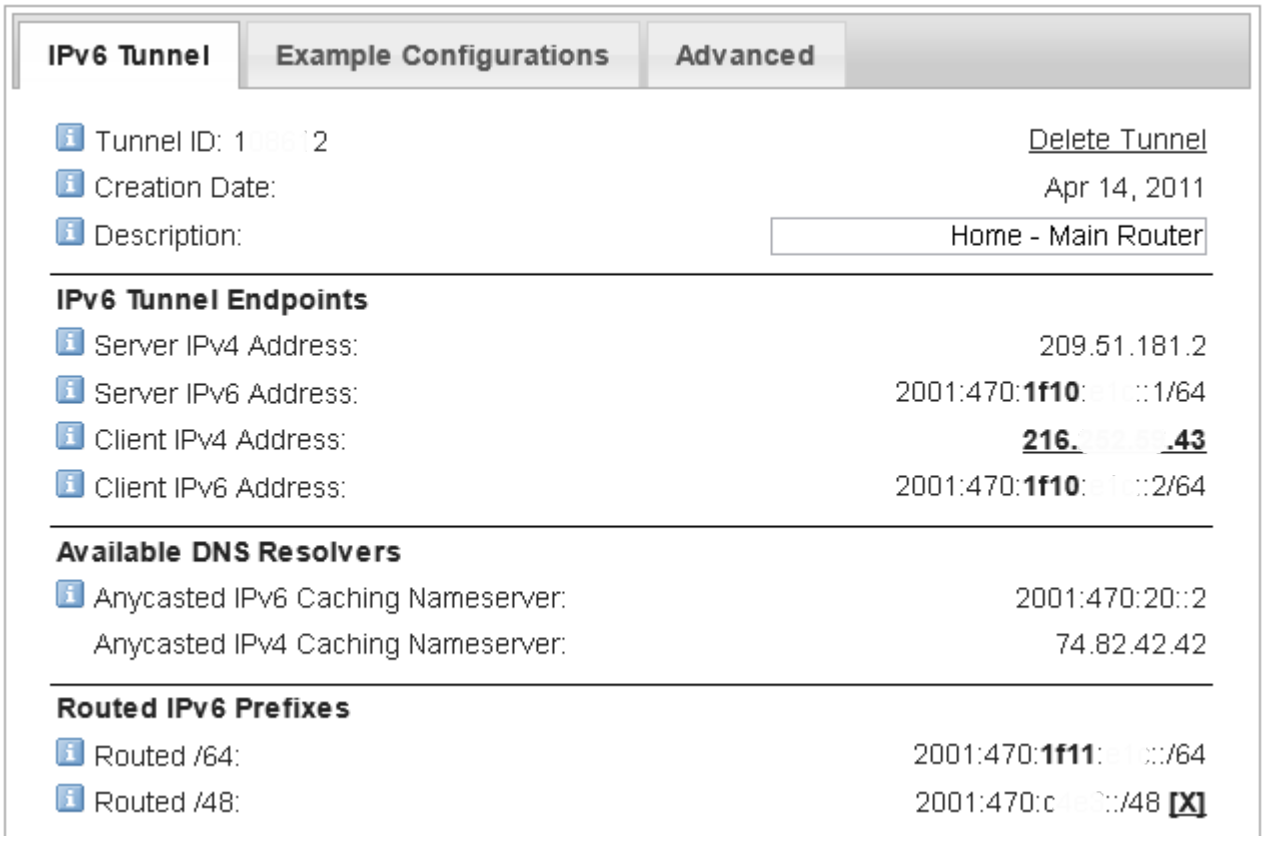
The most common way for a user that doesn't have access to native IPv6 connectivity to obtain service is by using a tunnel broker service such as Hurricane Electric's tunnelbroker service [http://www.tunnelbroker.net] or the similar service from SixXS [http://www.sixxs.net]. If you have a core site with IPv6 but not a remote site, it's also possible to deliver connectivity yourself with OpenVPN or a GIF tunnel or similar.

In this section, we will go over how to setup pfSense 2.1 to connect to Hurricane Electric (Often abbreviated to HE.net or HE) for IPv6 transit. Using HE.net is simple and easy, and they will let you setup multiple tunnels, each with a transport /64 and a routed /64. They'll even give you one routed /48 to be used with one of your tunnels. It's a great way to get a lot of routed IP space to experiment with and learn, all for free.

## Sign Up for Service

To get started, first you sign up with HE.net's service at http://www.tunnelbroker.net. Once you have registered and chosen a regional IPv6 tunnel server, you will be allocated your /64 networks. On the tunnel broker site, you can view a summary of your tunnel configuration like the one seen in Figure 2.3, “HE.net Tunnel Config Summary”. On that screen you'll see some important information such as your Tunnel ID, Server IPv4 Address (IP address of the tunnel server), Client IPv4 Address (your firewall's external IP address), the Server and Client IPv6 Addresses, that represent your IPv6 addresses inside the tunnel, and your Routed IPv6 Prefixes.

**Figure 2.3. HE.net Tunnel Config Summary**




Now that you have done the initial setup for the tunnel service, you can proceed to setting up pfSense to use the tunnel.

## Allow IPv6 Traffic

Navigate to System → Advanced on the Networking tab. Check the Allow IPv6 option if it is not already checked, then Save.

## Allow ICMP

Tunnel brokers, such a HE.net, often require you to allow ICMP to your WAN address that is terminating the tunnel in order to ensure you are online and reachable. If you block ICMP, the tunnel broker will often refuse to setup the tunnel to your IP. You can make a rule with the source IP of the Server IPv4 Address in your tunnel configuration, like the rule seen in Figure 2.4, “Example ICMP Rule”.

**Figure 2.4. Example ICMP Rule**


## Create and Assign the GIF Interface

Now you are ready to create the GIF interface tunnel. In pfSense, go to Interfaces → (assign) on the GIF tab. From here, add a new entry and fill in the fields on the screen with the corresponding information from the tunnel broker configuration.

The Parent Interface will be the WAN where the tunnel terminates. The one with your IP on the tunnel broker's Client IPv4 Address.

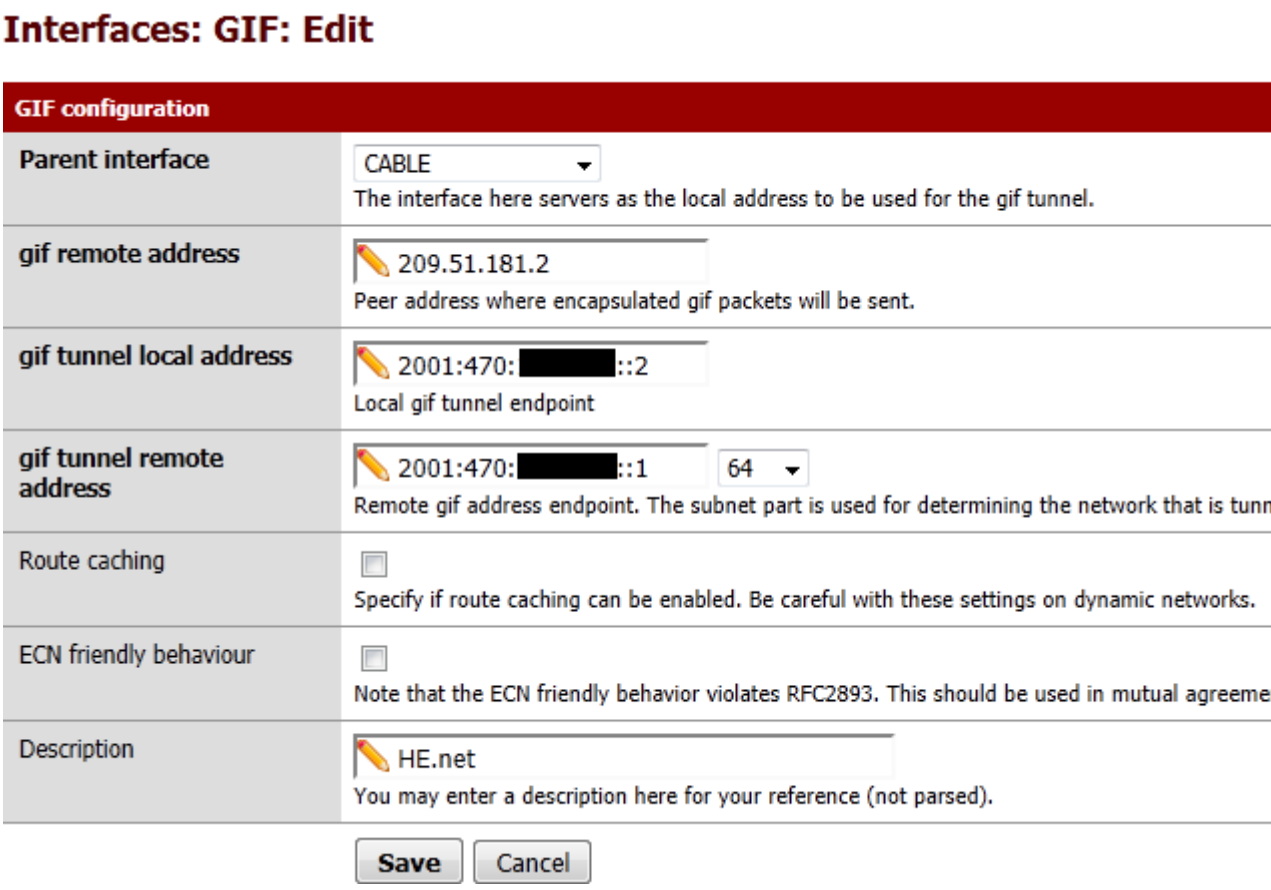
The GIF Remote Address in pfSense corresponds to the Server IPv4 Address from the tunnel broker.

The GIF Tunnel Local Address in pfSense corresponds to the Client IPv6 Address from the tunnel broker.

The GIF Tunnel Remote Address in pfSense corresponds to the Server IPv6 Address from the tunnel broker, along the with prefix length (typically /**64**).

The remaining options may be left blank or unchecked, and you can enter a Description if desired, then click Save. The end result looks something like seen in Figure 2.5, “Example GIF Tunnel”.

**Figure 2.5. Example GIF Tunnel**




If this tunnel is being configured on a WAN with a dynamic IP, see the section called “Updating the Tunnel Endpoint” for information on how to keep the tunnel's endpoint IP up to date with HE.net.

Now that the GIF tunnel has been created, it must be assigned. Go back to Interfaces → (assign) and click to add a new interface. Select the newly created GIF interface, which should show up in the list with the remote endpoint IP and description you just entered, then press Save.

## Configure the New OPT Interface

The new interface will now show up under Interfaces → OPTx, where *x* depends on what number interface this happened to be. Navigate there, and check Enable Interface.

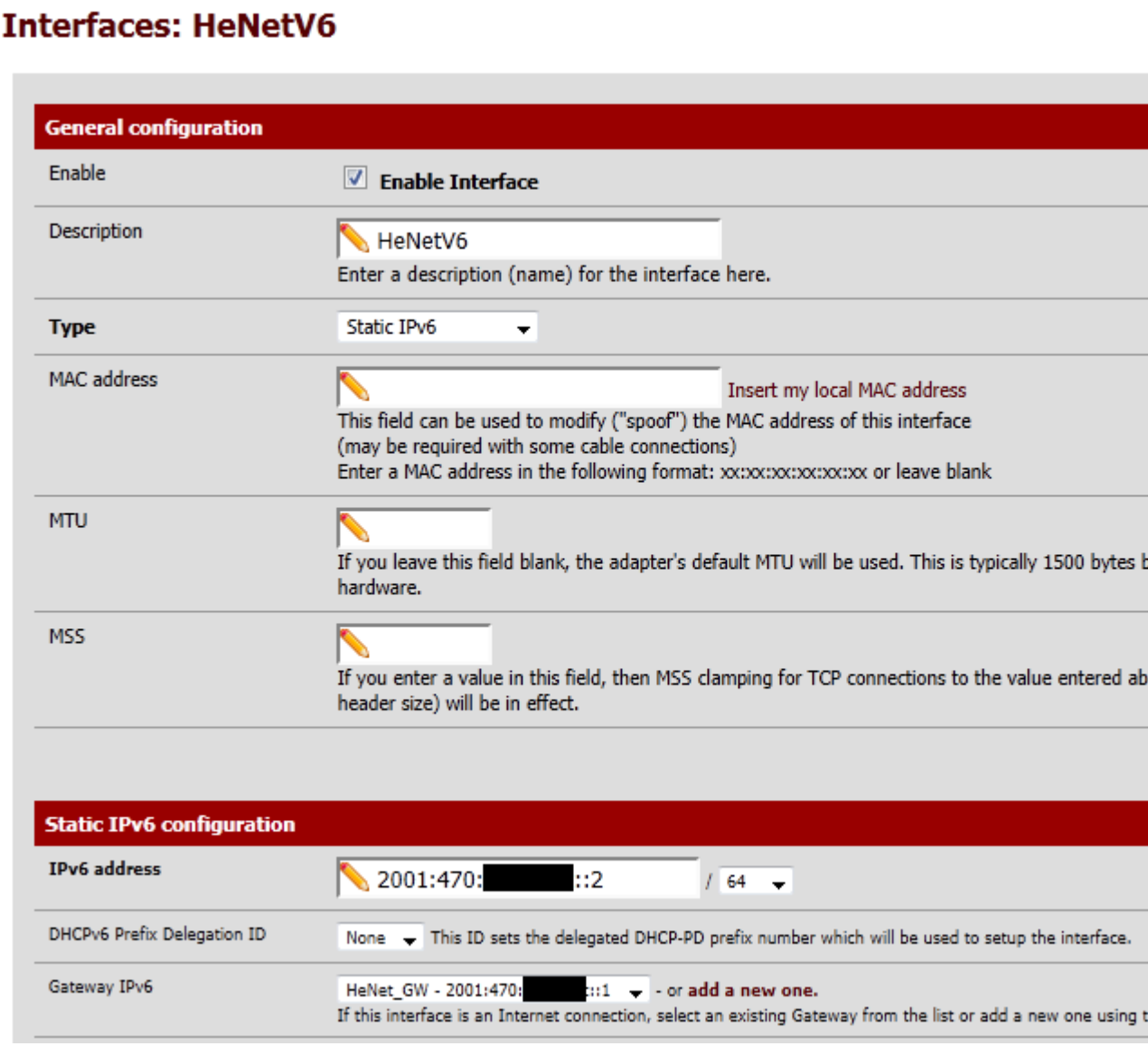
You can give the interface a better name by changing the Description field to something such as *HENETV6*. Then set the IPv6 Configuration Type to Static IPv6. Then, enter your Client IPv6 Address as the IPv6 Address, and use /**64** as the prefix length.

### Note

Some users have reported that their tunnel did not work properly with a prefix length of 64, but did work with a prefix length of 128 selected here. If you have issues passing traffic on your tunnel, you might try adjusting the prefix length accordingly.

The interface configuration will end up looking like Figure 2.6, “Example Tunnel Interface”.

**Figure 2.6. Example Tunnel Interface**

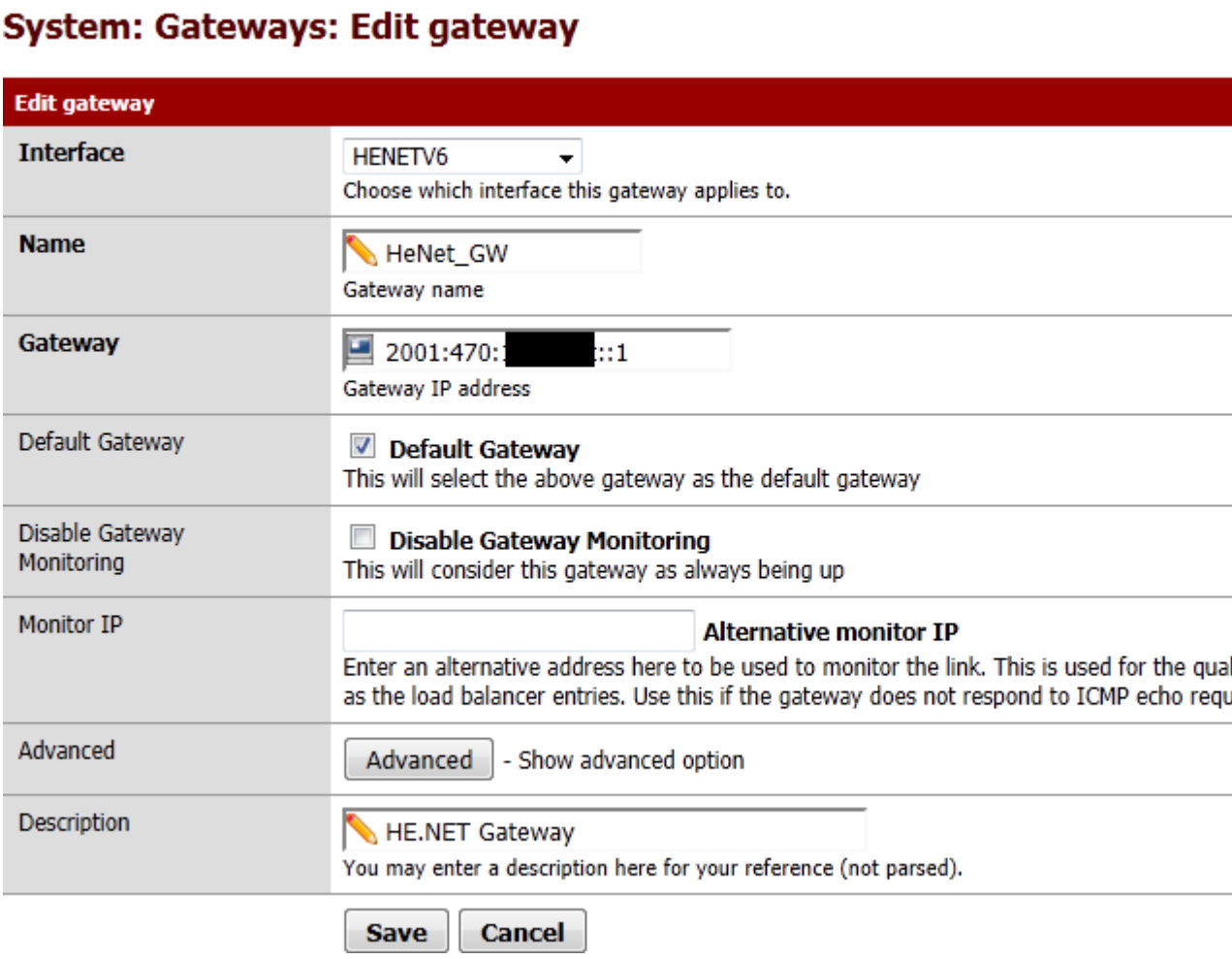



## Setup the IPv6 Gateway

The IPv6 gateway may be setup on the interface page by using the "add a new one" link, or by visiting System → Routing and adding the gateway there, then going back to the interface configuration page and selecting it there.

When creating the gateway, make sure that it is set to the correct interface, and that you use the Server IPv6 Address from your tunnel broker configuration. If this is your only, or your primary IPv6 gateway, ensure that Default Gateway is checked.

**Figure 2.7. Example Tunnel Gateway**




The gateway will look like the one shown in Figure 2.7, “Example Tunnel Gateway”, and when checked under Status → Gateways, it should show online as seen in Figure 2.8, “Example Tunnel Gateway Status”.

**Figure 2.8. Example Tunnel Gateway Status**


## Setup IPv6 DNS

Your current IPv6 servers likely already return AAAA results to get IPv6 answers for DNS queries, but you may want to enter the DNS servers supplied by your tunnel broker under System → General Setup. At least enter one IPv6 DNS server from there, or you can also use Google's public IPv6 DNS servers at 2001:4860:4860::8888 and 2001:4860:4860::8844.

## Setup LAN for IPv6

Now that the tunnel is configured and online, your firewall should have IPv6 connectivity from itself. However, that's only part of the battle, as you need to get IPv6 on your LAN as well, so clients can get out to the Internet on IPv6.

You can setup your LAN to be dual stack IPv4 and IPv6. Under Interfaces → LAN, select IPv6 Configuration Type to be Static IPv6 and then enter an IP address from your Routed /64 in the tunnel broker configuration, and select a prefix length of **64**.. Typically you want to use something such as *2001:db8:1111:2222::1* for the LAN IPv6 address if your Routed /64 is *2001:db8:1111:2222::/64*.

You can also use a /64 from somewhere inside your Routed /48 if you like, it's entirely up to you, but for simplicity's sake, this example uses Routed /64.

## Setup DHCPv6 and/or Router Advertisements

If you want to assign IPv6 addresses to clients automatically, then you'll need to setup Router Advertisements and/or DHCPv6. This is covered in great detail in the section called “IPv6 DHCP Server and Router Advertisements”.

Very briefly, you'll want to visit Services → DHCPv6 Server/RA, enable the server, enter a range of IPv6 IPs inside of your new LAN IPv6 subnet, click Save. Then switch to the Router Advertisements tab, set it for **Managed** or **Assisted**, and click Save there. For more details on what those modes mean, see the section called “Router Advertisements (Or: "Where is the DHCPv6 gateway option#")”.

If you wish to assign IPv6 addresses to your LAN systems manually, then set them up and ensure you use the firewall's LAN IP as their gateway, and use a proper matching prefix length.

## Add Firewall Rules

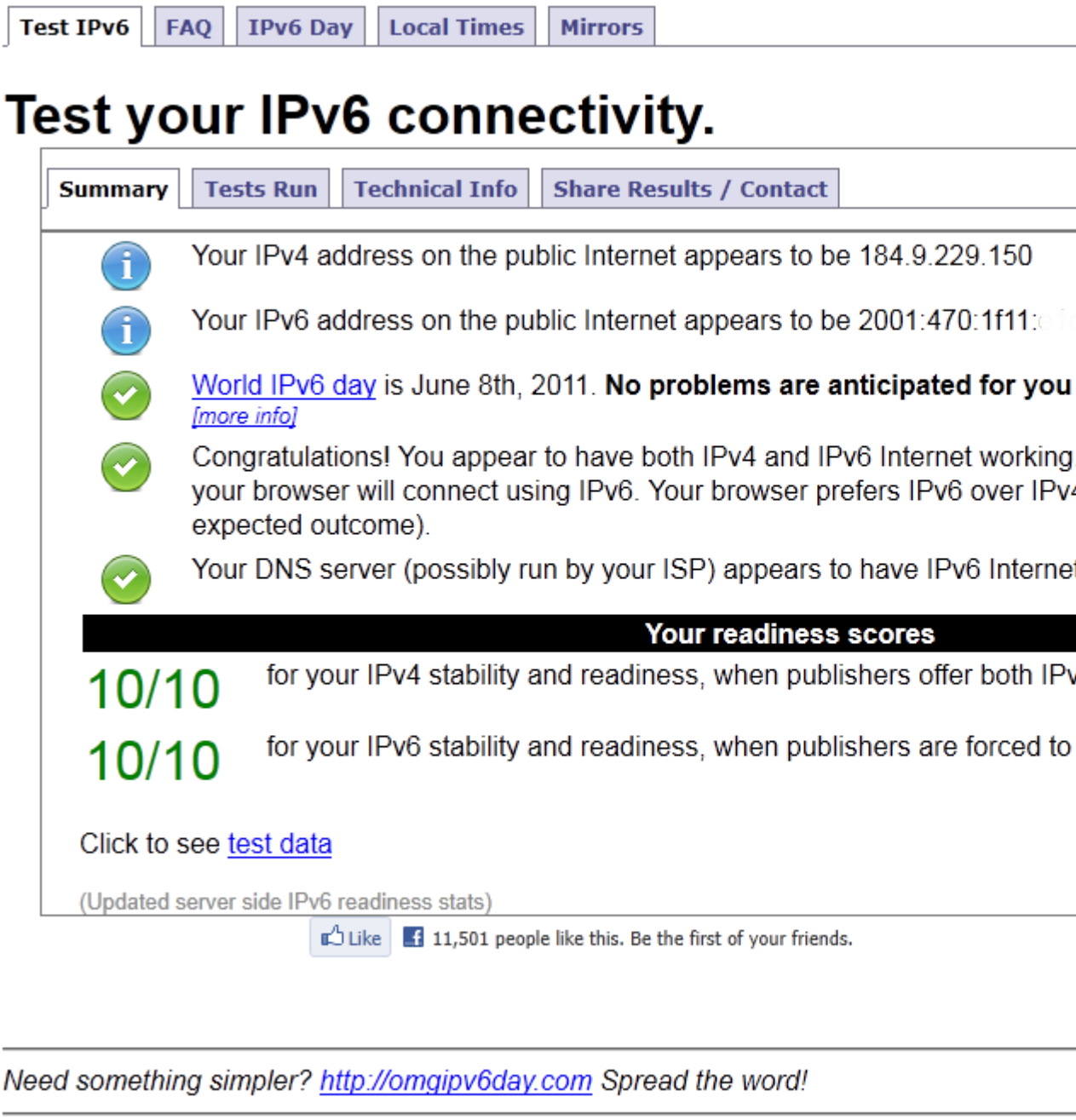
Now that your LAN addresses have been assigned, you need to add firewall rules to allow the IPv6 traffic to flow. To do this, visit Firewall → Rules and go to the LAN tab. Add a rule there with the TCP/IP Version set to IPv6, and enter the LAN IPv6 subnet as the source, with a destination of **Any**.

If you have IPv6-enabled servers on the LAN with public services, you can also add firewall rules on the tab for the IPv6 WAN (the assigned GIF interface) to allow IPv6 traffic to reach your servers on the ports that have public services.

## Try It!

Now that the firewall rules are in place, you should be able to get an IPv6 address on a LAN system and see if you have IPv6 connectivity. A good site to test with is http://test-ipv6.com/. If everything went as expected, you should see output like Figure 2.9, “IPv6 Test Results”.

**Figure 2.9. IPv6 Test Results**




And that's all! If the test succeeded, you now have IPv6 connectivity on your LAN to the rest of the world.

## Updating the Tunnel Endpoint

If you have a dynamic WAN, such as DHCP or PPPoE, you can still use HE.net as a tunnel broker. pfSense 2.1 includes a DynDNS type that will update your tunnel endpoint IP whenever your WAN interface IP changes.

To set that up, go to Services → DynDNS, click to add a new entry. Set the Service Type to be **`HE.net Tunnelbroker`**, and select your WAN as the Interface to Monitor. For the Hostname field, enter your Tunnel ID from the tunnel broker configuration. Then, enter your Username and Password for the tunnel broker site. Lastly, enter a Description if you like, and click Save.

# Controlling IPv6 Preference for traffic from the firewall itself

We do not yet have a way in the GUI to control whether services on the firewall itself will prefer IPv4 over IPv6. By default, it will prefer IPv6. This can be changed at the shell (or with a shellcmd tag.) You may need to change this in order to run a firmware upgrade or package upgrade remotely if for some reason your IPv6 routing is not functional, but the system believes it is up and working.

To prefer IPv4:

```
# env ip6addrctl_enable="yes" ip6addrctl_policy="prefer_ipv6" /etc/rc.d/ip6addr
```

To prefer IPv6 again:

```
# ip6addrctl flush
```

There will be a place in the GUI to control this in pfSense 2.2.

# Chapter 3. Hardware

pfSense is compatible with any hardware that is supported by the FreeBSD version in use, on i386 and amd64 hardware platforms. Alternate hardware architectures such as PowerPC, MIPS, ARM, SPARC, etc. are not supported at this time. The new embedded may bring MIPS and ARM support sometime in the future, though it is not available at the time of this writing. Starting with pfSense 2.0, there is now both a 64 bit release (amd64) and 32 bit release (i386). Older versions were 32-bit only, though the 32 bit release runs fine on 64 bit hardware. See the section called “32 bit vs. 64 bit” for more information on 32 bit vs 64 bit architectures.

# Hardware Compatibility

The best resource for determining compatible hardware is the FreeBSD Hardware Notes for the release version used by the pfSense release you are installing. pfSense 2.1 is based on FreeBSD 8.3, therefore a definitive reference on compatible hardware would be the hardware notes at http://www.freebsd.org/releases/8.3R/hardware.html. The more general FreeBSD hardware FAQ is another good resource to use for helping hardware selection. It can be found at http://www.freebsd.org/doc/en_US.ISO8859-1/books/faq/hardware.html. This section will provide guidance on the best supported hardware available for purposes of firewalling and routing. The primary consideration and only recommendation outside of the hardware notes is for network adapters.

## Network Adapters

Virtually all wired Ethernet cards (NICs) are supported by pfSense. However, not all network adapters are created equal. The hardware used can vary greatly in quality from one manufacturer to another, and in some cases, while FreeBSD may support a particular NIC, the driver support may be poor with a specific implementation of the chipset.

Intel Pro/100 and Pro/1000 NICs are the most commonly recommended because they have solid driver support in FreeBSD written by Intel employees, and perform well. On the other end of the spectrum, Realtek 8139 `rl` cards are extremely common but very poor quality hardware. A snippet of a comment in the source code for this driver tells the story — "The RealTek 8139 PCI NIC redefines the meaning of 'low end.' This is probably the worst PCI Ethernet controller ever made, with the possible exception of the FEAST chip made by SMC." Exacerbating the issue is the fact that numerous manufacturers incorporate this chipset in their NICs, with widely varying degrees of quality. You will find 8139 cards built into some embedded hardware, and those generally are reliable and function properly. Of the various PCI cards that exist, some work fine, and some have various things that are broken. VLANs may not work properly or at all, and promiscuous mode required for bridging may not work, amongst many other possibilities.

If you have NICs available and are building a system from spare parts, it is worthwhile to try what you have on hand. Many times they will work fine. If you are looking to buy hardware for your deployment, go with Intel cards. In networks where reliability and performance are of the utmost concern, don't skimp on costs by using whatever NICs you happen to have lying around (unless those happen to be Intel cards).

If using VLANs, ensure you select adapters that support VLAN processing in hardware. This is discussed in Chapter 14, *Virtual LANs (VLANs)*.

### USB Network Adapters

Many USB network adapters are supported, but generally not recommended. They perform poorly, especially on systems that do not support USB 2.0, or with adapters that are strictly USB 1.1. USB NICs are great in a pinch, or when adding network connectivity to a desktop PC, and are fine for some home firewall deployments, but for reliable performance in serious networks, they should not be considered.

### Wireless Adapters

Supported wireless adapters and recommendations are covered in the section called “Wireless drivers included in 2.1”.

# Minimum Hardware Requirements

The following outlines the minimum hardware requirements for pfSense 2.1. Note the minimum requirements are not suitable for all environments; see the section called “Hardware Sizing Guidance” for hardware sizing guidance.

## Base Requirements

The following requirements are common to all the pfSense platforms.

- CPU — 100 MHz or faster
- RAM — 256 MB or more, though 128 MB may suffice for the most basic of uses

## Platform-Specific Requirements

Requirements specific to individual platforms follow.

### Live CD

- CD-ROM, DVD, or other optical drive capable of booting a CD-ROM
- USB flash drive or floppy drive to store configuration file

### Full installation

- CD-ROM or bootable USB drive for initial installation
- 1 GB or larger hard drive

### NanoBSD Embedded

- 512 MB or larger Compact Flash card

As of pfSense 2.0.1 we also provide NanoBSD images for systems with VGA consoles, so the serial console is no longer a requirement for systems capable of using the NanoBSD VGA images.

# Hardware Selection

Open source operating systems can induce numerous headaches with hardware compatibility. While a particular piece of hardware may be supported, a specific implementation of it may not function properly, or certain combinations of hardware may not work. This isn't limited to FreeBSD (and hence pfSense) — Linux distributions also suffer the same fate. In more than a decade of experience using BSD and various Linux distributions on a wide variety of hardware, we have seen this countless times. Some systems that work fine with Windows won't work at all with BSD or Linux, some work fine with BSD but not Linux, some with Linux but not BSD. If you happen to run into hardware related problems, the section called “Hardware Troubleshooting” offers tips that will solve these issues in some instances.

## Preventing hardware headaches

This section offers some tips on avoiding hardware troubles.

## Use pfSense Certified Hardware

For best results, we highly recommend users stick with pfSense Certified hardware platforms. We developed our certified hardware program to provide assurance to the community that specific hardware platforms have been throughly tested and validated. Visit http://www.pfsense.org/hardware for the most up to date information on certified hardware.

## Search for the experiences of others

If you are using a piece of hardware from a major manufacturer, if you type its make, model, and **`site:pfsense.org`** into Google, there is a high probability you will find someone who has tried or is using that hardware. You also may want to try searching for the make, model, and **`pfSense`** to find experiences people have reported on other websites or the mailing list archives. Reports of failure shouldn't necessarily be considered definitive, as a single user's problems on a particular system could be the result of defective hardware or another anomaly rather than incompatibility. Repeating these same searches with **`FreeBSD`** instead of pfSense may also turn up useful user experiences.

## 32 bit vs. 64 bit

The main benefit that a 64-bit version offers in relation to firewalling is the ability to address more memory, and although even most of the largest pfSense installs protecting thousands of machines do not use 4 GB RAM with the base system, if you are using add-on packages, or require upwards of 3 million active states, the extra RAM may be needed. The 32 bit version of pfSense runs on 64 bit hardware, so it is a safe place to start if there is any uncertainty. We recommend sticking with 64 bit where the hardware supports it.

### Naming Conventions

Throughout this book we may refer to the 64 bit version as amd64, which is the designation used by FreeBSD for the architecture. Intel adopted the architecture created by AMD for x86-64, thus the name amd64, which refers to all x86 64 bit CPUs. The 32 bit platform is referred to as i386, which is also the designation used by FreeBSD for its 32 bit architecture.

# Hardware Sizing Guidance

When sizing hardware for use with pfSense, two main factors need to be considered: throughput required and features that will be used. The coming sections cover these considerations.

# Throughput Considerations

If you require less than 10 Mbps of throughput, you can get by with the minimum requirements. For higher throughput requirements we recommend following these guidelines, based on our extensive testing and deployment experience. These guidelines offer a bit of breathing room because you never want to run your hardware to its full capacity for extended periods.

Your choice of network card has a significant impact on the maximum achievable throughput, depending on the speed of your CPU. Table 3.1, “Maximum Throughput by CPU” shows the maximum achievable throughput using two Realtek 8139 NICs compared to two Intel Pro/1000 GT Desktop NICs for hardware platforms with PCI slots.

**Table 3.1. Maximum Throughput by CPU**

| CPU | Onboard Max Throughput (Mbps) | Realtek Max Throughput (Mbps) | Pro/1000 Max Throughput (Mbps) |
|---|---|---|---|
| Pentium MMX 200 MHz | n/a | 25 Mbps | 40 Mbps |

| CPU | Onboard Max Throughput (Mbps) | Realtek Max Throughput (Mbps) | Pro/1000 Max Throughput (Mbps) |
|---|---|---|---|
| WRAP — 266 MHz Geode | 24 Mbps | n/a | n/a |
| ALIX — 500 MHz Geode | 85 Mbps | n/a | n/a |
| VIA 1 GHz | 93 Mbps (100 Mb wire speed) | n/a | n/a |
| Netgate Hamakua (1 GHz Celeron) | 250 Mbps | n/a | n/a |
| Netgate FW-7535 (1.6 GHz Atom D510) | 485 Mbps | n/a | (onboard is Pro/1000) |
| Pentium II 350 MHz | n/a | 51 Mbps | 64 Mbps |
| Pentium III 700 MHz | n/a | 84 Mbps | 217 Mbps |
| Pentium 4 1.7 GHz | n/a | 93 Mbps (100 Mb wire speed) | 365 Mbps |

## Performance difference by network adapter type

Your choice of NIC will have a significant impact on performance. Cheap low end cards like Realteks will consume significantly more CPU than good quality cards such as Intel. Your first bottleneck with firewall throughput will be your CPU. You can get significantly more throughput out of a given CPU using a better quality NIC, as shown in Table 3.1, “Maximum Throughput by CPU” with the slower CPUs. If you have a CPU capable of significantly more throughput than you require, your choice of NICs will have little to no impact on throughput, though lesser quality NICs may prove unreliable in some circumstances.

## Sizing for gigabit throughput

When sizing for gigabit deployments, you first need to determine how much throughput you really need — 1 Gbps wire speed or just more than 100 Mbps. In many networks there are no systems capable of filling 1 Gbps with data from disk, as the systems' disk I/O is incapable of such performance. If you just want to be able to hit 200 Mbps, any 1 GHz system with good quality NICs will suffice. For up to 400-500 Mbps, an older 2-3 GHz server will suffice, or an Atom-based solution.

## Sizing for multiple gigabits per second deployments

The numbers in Table 3.1, “Maximum Throughput by CPU” stop at a relatively low level because that's the extent of what we can reasonably test in our lab. Testing multiple Gbps capable servers requires the servers and several systems capable of pushing 1 Gbps wire speed. We don't have adequate equipment for that scale of testing. But that's not to say that pfSense isn't suitable in such an environment; in fact, it's used in numerous deployments pushing in excess of 1 Gbps.

When sizing for multi-Gbps deployments, the primary factor is packets per second, not Gbps. You will hit the limit of FreeBSD and today's fastest quad core server hardware at around 500,000 packets per second (pps). How much throughput this will equate to depends on your network environment, with some references provided in Table 3.2, “500,000 pps throughput at various frame sizes”.

### Table 3.2. 500,000 pps throughput at various frame sizes

| Frame size | Throughput at 500Kpps |
|---|---|
| 64 bytes | 244 Mbps |

| Frame size | Throughput at 500Kpps |
|---|---|
| 500 bytes | 1.87 Gbps |
| 1000 bytes | 3.73 Gbps |
| 1500 bytes | 5.59 Gbps |

For deployments looking to achieve 1 Gbps wire speed between two interfaces, a Pentium 4 3 GHz or faster CPU with PCI-X or PCI-e NICs must be used. PCI will allow you to achieve several hundred Mbps, but PCI bus speed limitations will prevent you from achieving wire speed performance with two 1 Gbps NICs.

If you are sizing hardware for something capable of gigabit wire speed performance on multiple interfaces, get a new server with a quad core processor and PCI-e NICs and you will be in good shape. If you need to push more than 500,000 packets per second, you may exceed the capacity of commodity PC hardware to push packets. Refer to the section called “LAN Router” for more information.

# Feature Considerations

Most features do not factor into hardware sizing, though a few have significant impact on hardware utilization.

## Large State Tables

The firewall state table is where active network connections through the firewall are tracked, with each connection consuming one state. States are covered further in Chapter 10, *Firewall*. Environments requiring large numbers of simultaneous connections (and hence states) will require additional RAM. Each state takes approximately 1 KB of RAM. Table 3.3, “Large State Table RAM Consumption” provides a guideline for the amount of memory required for a large number of states. Keep in mind this is solely the memory used for the state tracking, the other components of pfSense will require at least 32-48 MB additional RAM on top of this and possibly more depending on the features in use.

### Table 3.3. Large State Table RAM Consumption

| States | RAM Required |
|---|---|
| 100,000 | ~97 MB |
| 500,000 | ~488 MB |
| 1,000,000 | ~976 MB |
| 3,000,000 | ~2900 MB |

## VPN (all types)

The question people usually ask about VPN is "how many connections can my hardware handle?" That is a secondary factor in most deployments, of lesser consideration. The primary consideration in hardware sizing for VPN is throughput required.

The encrypting and decrypting of network traffic with all types of VPNs is very CPU intensive. pfSense offers several cipher options for use with IPsec: DES, 3DES, CAST128, Blowfish 128-256 bit (in 8 bit increments), and AES 128, 192, and 256 bit. The various ciphers perform differently, and the maximum throughput of your firewall is dependent on the cipher used. 3DES is widely used because of its interoperability with nearly every IPsec device, however it is the slowest of all the ciphers supported by pfSense in absence of a hardware crypto accelerator. Hardware crypto accelerators such as supported cards from Hifn greatly increase maximum VPN throughput, and largely eliminate the performance difference between ciphers. Table 3.4, “IPsec Throughput by Cipher — ALIX” shows the maximum throughput by cipher for PC Engines ALIX hardware (500 MHz Geode) without and with a Soekris vpn1411 Hifn crypto accelerator.

### Table 3.4. IPsec Throughput by Cipher — ALIX

| Encryption Protocol | Maximum Throughput | Maximum Throughput (with Hifn) |
|---|---|---|
| DES | 13.7 Mbps | 34.6 Mbps |
| 3DES | 8.4 Mbps | 34.3 Mbps |
| Blowfish | 16.5 Mbps | not accelerated (no change) |
| CAST128 | 16.3 Mbps | not accelerated (no change) |
| AES | 19.4 Mbps | 34.2 Mbps |
| AES 256 | 13.5 Mbps | 34.2 Mbps |

Table 3.5, “IPsec Throughput by CPU” shows the maximum IPsec throughput by CPU for the Blowfish cipher at 128 bit, to illustrate maximum throughput capacity of various CPUs.

### Table 3.5. IPsec Throughput by CPU

| CPU | Blowfish Throughput (Mbps) |
|---|---|
| Pentium II 350 | 12.4 Mbps |
| ALIX (500 MHz) | 16.5 Mbps |
| Pentium III 700 | 32.9 Mbps |
| Pentium 4 1.7 GHz | 53.9 Mbps |

Hardware crypto accelerators should be used where high bandwidth through IPsec is required, except with dual or quad core CPUs, as those CPUs perform crypto faster than an accelerator by avoiding communicating on the PCI bus. Some VIA chipsets also support the VIA padlock crypto accelerator, which can also help. In future pfSense versions, the firewall will be able to take advantage of AES-NI which should dramatically improve the encryption performance of AES on supported chipsets, namely Intel's Core i5 and i7 line. Support for AES-NI was not included in FreeBSD 8.1, so it wasn't possible for pfSense 2.0 to include the feature.

## Packages

Some packages have a significant impact on the hardware requirements in your environment.

### Snort

Snort, the network intrusion detection system available in the pfSense package system, can require a significant amount of RAM, depending on your configuration. 256 MB should be considered a minimum, and some configurations may need 1 GB or more.

### Squid

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

In small environments, even for Squid, any hard drive will suffice. For 200+ user deployments using Squid, you should consider 10K RPM SATA or SCSI disks. Use 15K RPM SCSI or SAS disks for best performance in large environments.

pfSense supports most hardware RAID controllers found in server hardware. The use of RAID 10 on your RAID arrays can further improve Squid performance, and would be recommended for deployments with thousands of users.

# Hardware Tuning and Troubleshooting

The underlying OS beneath pfSense, can be fine-tuned in many ways. Some of these are available in pfSense under Advanced Options (See the section called “System Tunables Tab”), others are outlined in the FreeBSD man page tuning(7) [http://www.freebsd.org/cgi/man.cgi?query=tuning&apropos=0&sektion=0&manpath=FreeBSD+8.3-RELEASE&arch=default&format=html].

By default, we include a well-rounded set of values that are tuned for good performance without being overly aggressive. Still, there are cases where hardware or drivers will necessitate changing some values, or a specific network workload will require some changes to perform optimally. On our documentation wiki, we have a document outlining some of these common changes at http://doc.pfsense.org/index.php/Tuning_and_Troubleshooting_Network_Cards.

The most common problem encountered by users is mbuf exhaustion, which can lead to a kernel panic and reboot under certain network loads that exhaust all available network memory buffers. This is more common with some NICs than others, and mbuf usage also increases when some features, such as Limiters (the section called “Limiters”), are in use. To increase the amount of mbufs available, add the following to `/boot/loader.conf.local`:

```
kern.ipc.nmbclusters="131072"
```

Another common issue is that a NIC may not properly support MSIX, even though it claims to work. MSIX can be disabled by adding the following line to `/boot/loader.conf.local`:

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

After changing values in `/boot/loader.conf.local`, a reboot is required to activate them.

# Chapter 4. Installing and Upgrading

The hardware has been chosen, along with the pfSense version and platform to be used. Now it is time to download the appropriate pfSense release and install it on the target device. After downloading the proper version, continue to the section that describes installing the platform that has been chosen: Full Install or Embedded. If something should go wrong during the process, see the section called "Installation Troubleshooting" later in the chapter.

In this chapter, we also talk about recovery installation methods and how to upgrade pfSense. Recovery installations (the section called "Recovery Installation") are ways to reinstall pfSense with an existing configuration, typically with minimal downtime. Upgrading pfSense (the section called "Upgrading an Existing Installation") will keep your system current, add new features, and fix bugs. Upgrading is a fairly painless process which can be accomplished in several different ways.

# Downloading pfSense

Browse to www.pfsense.org [http://www.pfsense.org] and click the **Downloads** link. On the Downloads page, click the link for new installations. This will lead to the mirror selection page. Pick a mirror geographically close to your location for best performance. Once a mirror has been selected, a directory listing will appear with the current pfSense release files for new installations.

For Live CD or full installations, you can download the `.iso` file or the memstick image which is the same as the Live CD, but formatted for writing to USB media. The 2.1 release file name is `pfSense-2.1-RELEASE-`*`arch`*`.iso.gz` or `pfSense-memstick-2.1-RELEASE-`*`arch`*`.img.gz`. If you are downloading the 32 bit version, *`arch`* will be i386; For the 64 bit version, *`arch`* will be amd64. There is also a MD5 file available by the same name, but ending in `.md5`, and a SHA256 file ending in `.sha256`. These files contain a hash of the ISO or img file, which can be used to ensure the download completed properly. There are also serial memstick images, which work like the regular memstick, but output to the serial console instead. These images are useful for embedded units with hard drives or SSDs that lack a VGA output, but can boot from USB.

For embedded installations, download the appropriately sized NanoBSD `.img.gz` file. The 2.1 release file name is `pfSense-2.1-RELEASE-`*`size`*`-`*`arch`*`-nanobsd.img.gz`, where *`size`* is one of 512M, 1G, 2G, or 4G, to reflect the size of storage media (CF card, USB flash, etc.) for which that image is intended (sizes are in M for megabyte and G for gigabyte), and *`arch`* will be either i386 or amd64 depending on whether you want the 32 bit or 64 bit version. Typically you would want to match the size of the image to the size of your storage medium (CF card, USB flash), but you can use a smaller size image on a device such as a 1G image on a 2G CF card. This file is a gzipped image. You need not extract the file, as the installation process described later in this chapter will handle that. There is also a variation of the NanoBSD image file for each size that is setup to use a VGA console instead of serial, those are named `pfSense-2.1-RELEASE-`*`size`*`-`*`arch`*`-nanobsd_vga.img.gz` following the same rules as above for *`size`* and *`arch`*.

If at any point in the installation something does not go as described, check the section called "Installation Troubleshooting".

# Verifying the integrity of the download

The accompanying MD5 and SHA256 files can be used to verify the download completed successfully, and that an official release is being used.

## Hash verification on Windows

Windows users may install HashTab [http://implbits.com/Products/HashTab.aspx] or a similar program to view MD5 or SHA256 hashes for any given file. With HashTab installed, right click on the downloaded file and there will be a File Hashes tab containing the MD5 hash, among others. The generated MD5 hash can be compared with the contents of the `.md5` file downloaded from the pfSense

website, which is viewable in any plain text editor such as Notepad. If you do not see a SHA256 hash, right click in the hash view and click Settings, then check the box for SHA256 and click OK.

## Hash verification on BSD and Linux

The **md5** command comes standard on FreeBSD, and many other UNIX and UNIX-like operating systems. An MD5 hash may be generated by running the following command from within the directory containing the downloaded file:

```
# md5 pfSense-2.1-LiveCD-Installer.iso
```

Compare the resulting hash with the contents of the `.md5` file downloaded from the pfSense website. GNU or Linux systems provide a **md5sum** command that works similarly. To verify using SHA256, the commands work similarly, you need only replace references to **md5** with **sha256** in the command and file names.

## Hash verification on OS X

OS X also includes the **md5** command just like FreeBSD, but there are also GUI applications available such as MD5 from Eternal Storms [http://www.eternalstorms.at/md5/]. HashTab [http://implbits.com/Products/HashTab.aspx] is also available for OSX.

# Full Installation

This section describes the process of installing pfSense to a hard drive. In a nutshell, this involves booting from the Live CD or memstick, performing some basic configuration, and then invoking the installer from the CD. If you encounter problems while trying to boot or install from CD, see the section called “Installation Troubleshooting” later in the chapter. If you want to use the memstick image instead of the Live CD, instructions for writing the image are the same as writing the embedded image covered in the section called “Embedded Installation”. After writing the memstick image, proceed to the section called “Booting the CD”.

### Note

If the target hardware does not have a CD-ROM drive and cannot boot from USB, a different machine may be used to install on the target hard drive. See Alternate Installation Techniques (the section called “Alternate Installation Techniques”) for more information.

# Preparing the CD

A CD will need to be burned from the ISO image downloaded in the previous section. Since the downloaded file is a CD image, it will need to be burned appropriately for image files — *not* as a data CD containing the single ISO file. Procedures for doing so will vary by OS and software available.

Before the image can be burned, it must be decompressed. The `.gz` extension on the file indicates that it's compressed with **gzip**. This can be decompressed on Windows using 7-Zip [http://www.7-zip.org/], or on BSD/Linux/Mac with the **gunzip** or **gzip -d** commands.

## Burning in Windows

Virtually every major CD burning software package for Windows includes the ability to burn ISO images. Refer to the documentation of the CD burning program being used. A Google search with the name of the burning software and "**`burn iso`**" should help to locate instructions.

### Burning with Nero

It is easy to burn ISO images with Nero. Start by right clicking on the ISO file, then click Open With, and select Nero. The first time this is done, it may be necessary to select Chose Default Program

and then pick Nero from the list. This same process should work with other commercial CD burning software.

### Burning with ISO Recorder

If using Windows XP, 2003, or Vista, the freely available ISO Recorder [http://isorecorder.alexfeinman.com] tool may be used. Download and install the appropriate version of ISO Recorder for the operating system being used, then browse to the folder on the drive containing the pfSense ISO, right click on it, and click Copy image to CD.

### Other Free Burning Software

Other free options for Windows users include CDBurnerXP [http://www.cdburnerxp.se/], InfraRecorder [http://infrarecorder.org/] and ImgBurn [http://www.imgburn.com/], among others. 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 some form of GUI CD burning application that can handle ISO images. If one is integrated with the window manager, right click on the ISO file and choose Write disc to. Other popular choices include K3B and Brasero Disc Burner.

If there is not a GUI burning program installed, it may still 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
Cdrecord-Clone 2.01 (i686-pc-linux-gnu) Copyright (C) 1995-2004 Jörg Schilling
Linux sg driver version: 3.1.25
Using libscg version 'schily-0.8'.
scsibus0:
   0,0,0 100) 'LITE-ON ' 'COMBO LTC-48161H' 'KH0F' Removable CD-ROM
```

Note the SCSI ID/LUN is *0,0,0*. Burn the image as in the following example, replacing ***<max speed>*** with the speed of the burner and *lun* with the SCSI ID/LUN of your recorder:

```
# cdrecord --dev=lun --speed=<max speed> \
    pfSense-2.0-RELEASE-i386-LiveCD-Installer.iso
```

## Burning in FreeBSD

FreeBSD includes the burncd program in its base system which can be used to burn ISO images like so.

```
# burncd -s max -e data pfSense-2.0-RELEASE-i386-LiveCD-Installer.iso fixate
```

For more information on creating CDs in FreeBSD, please see the entry for CD burning in the *FreeBSD Handbook* at http://www.freebsd.org/doc/en/books/handbook/creating-cds.html.

## Verifying the CD

Now that the CD is prepared, verify it was burned properly by viewing the files contained on the CD. More than 20 folders should be visible, including bin, boot, cf, conf, and more. If only one large ISO file is seen, the CD was not burned properly. Repeat the steps listed earlier for burning a CD, and be sure to burn the ISO file as a CD image and not as a data file.

# Booting the CD

Now power on the target system and place the CD into the drive. pfSense should begin to boot, and will show an assign interfaces prompt which is covered in a following section.

### Note

If you are using a USB optical drive you should press option **3** at the boot menu, Boot pfSense using USB device. This introduces a 10 second delay into the boot process which allows full detection of USB devices before the boot proceeds. Without this option, many systems booting will fail at a `mountroot>` prompt. This is done automatically when booting the USB memstick image. For other boot issues, see the section called "Installation Troubleshooting" later in the chapter.

## Specifying Boot Order in BIOS

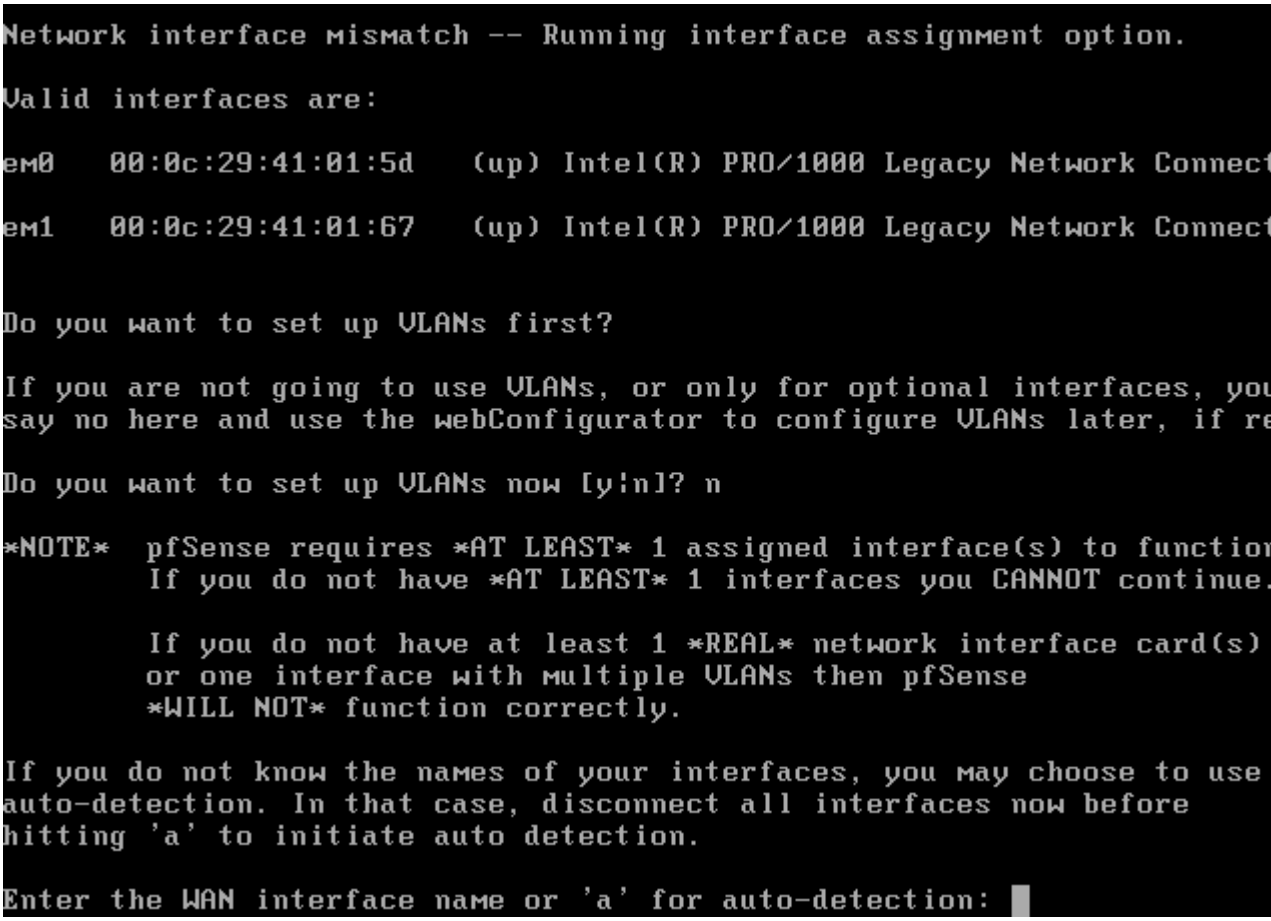
If the target system did not boot from the CD or the USB memstick, the most likely reason is that the given device was not early enough in the list of boot media in the BIOS. Many newer motherboards also allow bringing up a one time boot menu by pressing a key during POST, commonly **Esc** or **F12**.

Failing that, change the boot order in the BIOS. First, power on the system and enter the BIOS setup. It is typically found under a Boot or Boot Priority heading, but could be anywhere. If booting from CD-ROM or USB is not enabled, or has a lower priority than booting from the hard drive and the drive contains another OS, the system will not boot from the pfSense media. Consult the motherboard manual for more detailed information on altering the boot order.

# Assigning Interfaces

After the pfSense Live CD has completed the boot process, the system will prompt for interface assignment as in Figure 4.1, "Interface Assignment Screen". This is where the network cards installed in the system are given their roles as WAN, LAN, and Optional interfaces (OPT1, OPT2 ... OPTn).

**Figure 4.1. Interface Assignment Screen**



A list of network interfaces and their MAC addresses that were located on the system will appear, 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. The MAC (Media Access Control) address of a network card is a unique identifier assigned to each card, and no two network cards should have the same MAC address. (In practice, this is not quite true, MAC address duplication does happen fairly often.) After that, a prompt will show up for VLAN configuration. If VLANs are desired, see Chapter 14, *Virtual LANs (VLANs)* later in the book for details of their setup and usage. Otherwise, type **n** and press enter.

The WAN interface is configured first. As you will often have more than one network card, a dilemma may present itself: How to tell which is which? If the identity of each card is already known, simply enter the proper device names for each interface. If the difference between network cards is unknown, the easiest way to figure it out would be to use the auto-detection feature.

For automatic interface assignment, first unplug all network cables from the system, then type **a** and press enter. Now plug a network cable into the interface that should connect to the WAN, and press enter. If all went well, pfSense should know now which interface to use for the WAN. The same process may be repeated for the LAN, and any optional interfaces that will be needed. If a message is displayed such as No link-up detected, see the the section called “Installation Troubleshooting” for more information on sorting out network card identities.

After the interfaces you want have been configured, press enter when asked for another interface, and a prompt will appear asking `Do you want to proceed?`. If the network interface assignment appears correct, type **y**, then press enter. If the assignment is not right, type **n** and press enter to repeat this process.

### Note

pfSense 2.0 and later support only assigning a single interface. This is called Appliance Mode. In this mode, the system will move the GUI anti-lockout rule to the WAN interface so you may access the firewall from there. The usual routing functions would not be active since there is no "internal" interface. This type of configuration is useful for VPN appliances, DNS servers, etc.

# Installing to the Hard Drive

Once the interface assignment is complete, a menu will appear with additional tasks that may be performed. To install pfSense to the hard drive of this system, choose option **99** which will launch the installation process.

The first screen to appear will ask to adjust console settings. Unless an alternate language keyboard is being used, choose Accept These Settings and move on to the next step.

Next, a list of tasks will be presented. If there is only one hard drive installed on the system and you do not need to set any other custom options, Quick/Easy Install may be chosen. This will install to the first hard drive it finds and accept all of the default options. A confirmation dialog will be shown. Press OK to continue or Cancel to return to the previous menu. The installation will proceed and only stop to prompt for which kernel should be installed.

If you chose to use the Quick/Easy Install option, skip ahead to Table 4.1, “Kernel Choices” for kernel choices. Otherwise, pick the first option: Install pfSense to perform a custom installation and continue on through the rest of this section.

Now pick the hard drive to which pfSense will be installed. Each hard drive attached to the system should be shown, along with any supported RAID or gmirror volumes. Select the drive with the up and down arrows, then press enter. If no drives are found or the incorrect drives are shown, 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 the section called “Installation Troubleshooting” for help.

The next step is to format the drive that was just chosen. Unless it is known for certain that the drive contains a usable FreeBSD partition, select Format This Disk and press enter. Otherwise, choose Skip this step. When presented with the Disk Geometry screen, it is best to choose Use this geometry. It is possible to override this if more correct values are known, but in most cases the defaults are correct. A confirmation screen will be shown, at which point the Format <drive name> option must be chosen to continue.

## Note

This is a good place to stop and ensure that the correct drive has been selected, as there is no turning back once this action has been performed. *Everything on the disk will be destroyed*.

Dual booting with another operating system is possible for advanced users who know how to manually configure such things, but such configurations are not officially supported and will not be detailed here.

Partitioning follows, and you should simply accept the defaults by choosing Accept and Create, then choosing Yes, partition at the next screen.

A prompt is then shown for installing bootblocks. This is what will allow the hard drive to boot. Install Bootblocks will already be selected (an X appears in that column next to the drive being configured). Packet Mode may or may not be needed, depending on the hardware combination in use. Some newer hardware and larger drives will work better with packet mode enabled, and older hardware may prefer packet mode disabled. Leave the defaults selected unless they do not work on your system for some reason. Now select Accept and Install Bootblocks and press enter. A confirmation box will appear with the result of that command, and if it succeeded, press enter one more time to proceed.

Select the partition on which to install pfSense at the next screen. If the defaults were used as suggested, there is likely only one choice. If multiple choices appear, pick the one that was created for pfSense. Another confirmation window will appear reporting the success of the formatting process.

Subpartitions may now be created, but again the defaults on this screen will be acceptable for nearly all uses. Some people prefer to have separate subpartitions for `/var`, `/tmp`, etc. but this is not necessary, and you should not do so unless you have a considerable understanding of the space requirements specific to your installation. If you are performing a full install to flash-based media like a CF card or USB thumb drive, be sure to remove the `swap` partition. Make the desired changes, and then select Accept and Create.

Now sit back, wait, and have a few sips of coffee while the installation process copies pfSense to the target location. After the installation process has finished its work, there is a final prompt to select which kernel to install on the target system. There are two options available, each with its own purpose:

### Table 4.1. Kernel Choices

| Kernel Type | Purpose/Description |
|---|---|
| Standard Kernel | Used for normal systems which have a VGA console. |
| Embedded Kernel | Disables VGA console and keyboard, uses serial console. |

The Standard Kernel will work regardless of the number of processors available, and is the current recommended default. Previously there had been a uniprocessor kernel choice, but thanks to advances in FreeBSD there is no longer a benefit to excluding SMP support from the default kernel. There also used to be a Developer kernel but that is also no longer needed in almost all cases, so it has been removed.

## Note

One of the reasons a debug kernel is no longer needed is because we can now automatically gather information about crashes using a regular kernel. The debug

kernel was taking up extra space, and offered little benefit. On full installs with swap space configured, should a kernel panic or similar crash happen, the firewall will now automatically collect information about the crash and then reboot. Once the system has recovered, you will find a prompt on the dashboard to view and submit the crash report to our servers, or delete the crash data without submitting it. When contacting support or using the forum/mailing list, the information contained in this crash report may be useful in diagnosing the problem.

When the installation is complete, select Reboot, and then once the system has restarted, remove the CD before the boot process begins.

Congratulations, pfSense is now fully installed!

# Embedded Installation

The embedded version is released as a disk image, which must be written out to a Compact Flash card (CF), USB flash drive, or other storage medium using **physdiskwrite** or **dd**. After the image is written, it is then placed in the target device and configured.

### Note

Be **very careful** when doing this!! If this is run this on a machine containing other hard drives it is possible to select the wrong drive and overwrite a portion of that drive with pfSense. This renders the disk completely unreadable except to certain disk recovery programs, and that is hit and miss at best. **physdiskwrite** for Windows contains a safety check that will not allow overwriting a drive larger than 800 MB without a specific option at the command line. The safest way to install pfSense to a CF is through USB redirection with VMware, discussed later in this chapter in the Alternate Installation Techniques section (the section called “Alternate Installation Techniques”).

## Embedded Installation in Windows

The **physdiskwrite** program by Manuel Kasper, author of m0n0wall, is the preferred means of writing the pfSense image to CF in Windows. It can be downloaded from the m0n0wall website [http://m0n0.ch/wall/physdiskwrite.php]. Save it somewhere on the PC being used, such as `C:\tools` or another convenient location. If another location is chosen, substitute `C:\tools` in the example with the directory where `physdiskwrite.exe` has been placed.

### Note

There is also a GUI available for **physdiskwrite** called PhysGUI, but the only available version as of this writing was in German. That said, the GUI is simple enough to use that it may not be a barrier for many people. In fact, it may prove easier to use, even in a foreign language, than the command line version is in English. For example, identifying the proper device is a much simpler task. It is also available from the m0n0wall website. There are other GUI-driven tools for imaging drives such as Image Writer for Windows [https://launchpad.net/win32-image-writer]. If you are more comfortable using a GUI, feel free to pursue other imaging methods. Be aware that some other tools may expect the downloaded image to be decompressed before writing.

In Windows Vista or Windows 7, **physdiskwrite** must be launched from a command prompt run as administrator. Simply having administrator rights is not enough. The easiest way to do this is to click the Start button, then type **`cmd`** in the search box. Right click on `cmd.exe` when it pops up and choose Run as Administrator. The **physdiskwrite** program may then be run from that command prompt without any problems. Running it from a command prompt which has not been run as administrator will result in no disks being found.

To use **physdiskwrite**, first start a command prompt.

Then change to the directory containing `physdiskwrite.exe` and run it followed by the path to the pfSense `.img.gz` file downloaded earlier. After running the command, a prompt with a list of drives attached to the system will appear. The safest way to ensure the correct drive is chosen would be to run **physdiskwrite** before inserting the CF, record the output, then press **Ctrl**+**C** to exit. Insert the CF and run **physdiskwrite** again, comparing the output to the previous run. The disk shown now that was not shown previously is the CF. The number of cylinders ("cyl" in **physdiskwrite** output) may also be used to help indicate the proper drive. The 512 MB CF used in the following example has 63 cylinders, while the hard drives all have over 30,000. Also remember that **physdiskwrite** has a safety mechanism that will not overwrite a disk larger than 2 GB without specifying `-u` after the **physdiskwrite** command.

After selecting the disk to write, **physdiskwrite** will write out the image. This will take between two to ten minutes on a fast machine with USB 2.0 and a USB 2.0 CF writer. If the system or CF writer is only USB 1.1, expect it to take several times longer due to the very low speed of USB 1.1. Larger images, such as the 4GB image, can take significantly longer on some systems.

The following is a practical example of using **physdiskwrite** to write a pfSense image.

```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Windows\system32> cd \tools

C:\tools> physdiskwrite.exe c:\temp\pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz

physdiskwrite v0.5.1 by Manuel Kasper <mk@neon1.net>

Searching for physical drives...

Information for \\.\PhysicalDrive0:
   Windows:       cyl: 36481
                  tpc: 255
                  spt: 63

Information for \\.\PhysicalDrive1:
   Windows:       cyl: 30401
                  tpc: 255
                  spt: 63

Information for \\.\PhysicalDrive2:
   Windows:       cyl: 63
                  tpc: 255
                  spt: 63

Information for \\.\PhysicalDrive3:
DeviceIoControl() failed on \\.\PhysicalDrive3.

Information for \\.\PhysicalDrive4:
DeviceIoControl() failed on \\.\PhysicalDrive4.

Information for \\.\PhysicalDrive5:
DeviceIoControl() failed on \\.\PhysicalDrive5.

Information for \\.\PhysicalDrive6:
   Windows:       cyl: 30515
                  tpc: 255
                  spt: 63

Information for \\.\PhysicalDrive7:
```

```
   Windows:       cyl: 0
                  tpc: 0
                  spt: 0

Which disk do you want to write? (0..7) 2
About to overwrite the contents of disk 2 with new data. Proceed? (y/n) y
Found compressed image file
122441728/122441728 bytes written in total

C:\tools>
```

After **physdiskwrite** has completed, the CF may be removed from the writer and placed in the target hardware.

### Note

The written CF contains BSD filesystem formatted partitions that are not readable in Windows. Windows will claim the drive needs to be formatted should you try to access it. Do not do so, simply move the CF to the target hardware. There is no way to view the contents of the written CF in Windows.

### Note

There is a bug in physdiskwrite with Windows 7 and newer that will cause it to fail with a write error shortly after it starts writing when the storage medium has a Windows-readable filesystem on it. Delete all the partitions from Disk Management, or open a command prompt as administrator and perform the following steps.

- Type **diskpart** and press enter.
- Type **list disk** and press enter to find out the number of your drive.
- Type **select disk X** (where you replace X with the number of your drive) and hit enter.
- Type **clean** and press enter.

Now continue to write the storage device as previously shown.

# Embedded Installation in Linux

Embedded installation in Linux is accomplished by piping the **gunzip** output from the image to **dd**.

```
# gunzip -c pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz | dd of=/dev/hdX bs=16k
```

where *X* specifies the IDE device name of the CF card or IDE disk (check with **hdparm -i /dev/hdX**) — some adapters, particularly USB, may show up under SCSI emulation as `/dev/sdX`.

*Ignore the warning about trailing garbage — it's because of the digital signature.*

# Embedded Installation in FreeBSD

**gzip** piped to **dd** will write the image out to CF in FreeBSD. Before starting, you will need to know the device name which corresponds to the CF card in use. If a hard drive or CF-to-IDE adapter is being used, it may be an `ad` device such as `ad0`. Check the output of **dmesg** or `/var/log/messages`. If a USB CF reader is being used, it may be a `da` device such as `da0`, check `/var/log/messages` after plugging in the card reader, it should report which device was added.

To image the card, you should be able to decompress the image and copy it to the card in one step:

```
# gzip -dc pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz | dd of=/dev/adX obs=64k
```

*Ignore the warning about trailing garbage — it's because of the digital signature.*

If the imaging stops short or errors off after only transferring a small amount of data, you may need to decompress the image first:

```
# gunzip pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz
# dd if=pfSense-2.1-RELEASE-2g-i386-nanobsd.img of=/dev/adX obs=64k
```

# Embedded Installation in Mac OS X

This process has been tested on Mac OS X 10.3.9 and later, up to and including Snow Leopard/10.6. It is recommended that you disconnect all disks except for your startup disk before carrying out this procedure, as an error in specifying the drive to be written to will cause data loss.

- Plug in your CF reader with CF card inserted.
- If Mac OS X pops up a message saying that the card could not be read, click Ignore.
- Open Disk Utility.
- Select any Partitions of your CF Card that are mounted, and click the unmount button. The partitions should now appear in grey.
- Select your CF Card Reader in the left-hand column, and click the Info button.
- Note the 'Disk Identifier': e.g. 'disk1'.
- Open Terminal.
- Change to the directory containing the pfSense image.
- Use this command, replacing *disk[n]* with the disk identifier found above:

  ```
  # gzcat pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz | dd of=/dev/disk[n] bs=16k
  ```

There is also the following alternative to accomplish this entirely from the command line.

```
$ diskutil list
/dev/disk0
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:   GUID_partition_scheme                        *298.1 Gi   disk0
   1:                        EFI                     200.0 Mi   disk0s1
   2:                  Apple_HFS Macintosh HD        297.8 Gi   disk0s2
/dev/disk1
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     CD_partition_scheme 30 Days To Great French *521.4 Mi   disk1
   1:                     CD_DA                       7.8 Mi     disk1s1
   2:                     CD_DA                       7.8 Mi     disk1s2
   3:                     CD_DA                       18.2 Mi    disk1s3
   4:                     CD_DA                       13.8 Mi    disk1s4
   5:                     CD_DA                       14.0 Mi    disk1s5
   6:                     CD_DA                       12.1 Mi    disk1s6
   7:                     CD_DA                       14.2 Mi    disk1s7
   8:                     CD_DA                       21.5 Mi    disk1s8
   9:                     CD_DA                       16.6 Mi    disk1s9
  10:                     CD_DA                       14.7 Mi    disk1s10
  11:                     CD_DA                       24.3 Mi    disk1s11
```

```
 12:                    CD_DA                      16.6 Mi    disk1s12
 13:                    CD_DA                      22.4 Mi    disk1s13
 14:                    CD_DA                      14.7 Mi    disk1s14
 15:                    CD_DA                      20.5 Mi    disk1s15
 16:                    CD_DA                      19.4 Mi    disk1s16
 17:                    CD_DA                      15.3 Mi    disk1s17
 18:                    CD_DA                      17.9 Mi    disk1s18
 19:                    CD_DA                      18.2 Mi    disk1s19
 20:                    CD_DA                      16.0 Mi    disk1s20
 21:                    CD_DA                      26.8 Mi    disk1s21
 22:                    CD_DA                      18.8 Mi    disk1s22
 23:                    CD_DA                      21.7 Mi    disk1s23
 24:                    CD_DA                      14.5 Mi    disk1s24
 25:                    CD_DA                      22.2 Mi    disk1s25
 26:                    CD_DA                      16.7 Mi    disk1s26
 27:                    CD_DA                      20.9 Mi    disk1s27
 28:                    CD_DA                      16.0 Mi    disk1s28
 29:                    CD_DA                      20.8 Mi    disk1s29
 30:                    CD_DA                      17.1 Mi    disk1s30
/dev/disk2
  #:                       TYPE NAME                    SIZE       IDENTIFIER
  0:   GUID_partition_scheme                        *90.0 Mi    disk2
  1:               Apple_HFS Processing              90.0 Mi    disk2s1
/dev/disk3
  #:                       TYPE NAME                    SIZE       IDENTIFIER
  0:  FDisk_partition_scheme                        *978.5 Mi   disk3
  1:              DOS_FAT_32 UNTITLED                978.4 Mi   disk3s1
$ diskutil umount disk3s1
$ gzcat pfSense-2.1-RELEASE-2g-i386-nanobsd.img.gz | dd of=/dev/disk3 bs=16k
7665+1 records in
7665+1 records out
125587456 bytes transferred in 188.525272 secs (666157 bytes/sec)
```

# Completing the Embedded Installation

Now that the storage device contains a pfSense image, it can be placed into the target device, but it may still need some configuration. Users of ALIX and Soekris 5501 hardware can skip this section, as they use vr(4)-based network controllers, and the default embedded installation assumes that vr0 is LAN and vr1 is WAN. These ports should be labeled on the hardware. If you want to reassign these interfaces from the console instead of the WebGUI, continue to the next section.

## Connect a Serial Cable

First, a null modem [http://en.wikipedia.org/wiki/Null_modem] serial cable should be connected between the device and a PC. Depending on the serial port and cable being used, a serial cable gender changer [http://en.wikipedia.org/wiki/Gender_changer] may also be necessary to match the available ports. If a real null modem serial cable is unavailable, there are also null modem adapters that will convert a standard serial cable into a null modem cable.

## Start a Serial Client

On the PC being used to configure the embedded device, a serial client program must be used. Some popular clients for Windows are Hyperterminal, which should be on almost any XP installation, and PuTTY [http://www.chiark.greenend.org.uk/~sgtatham/putty/], which is free and much more reliable. On Linux, `minicom` should be present in most distribution package systems. On FreeBSD, simply use the built-in program `tip`. Typing **`tip com1`** (Or **`tip ucom1`** if you are using a USB serial adapter) will connect to the first serial port. Disconnect by typing "**`~.`**" at the start of a line.

Whichever serial client is used, ensure that it is set for the proper Speed (9600), Data Bits (8), Parity (No), and Stop Bits (1). This is typically written as 9600/8/N/1. Some embedded units default to a faster speed. PC Engines WRAP and ALIX default to 38400/8/N/1 and Soekris hardware defaults to 19200/8/N/1. Many serial clients default to 9600/8/N/1, so adjusting these settings may not be necessary. You will need to use 9600/8/N/1 with pfSense regardless of the setting of your hardware. For hardware using speeds other than 9600, you will likely want to change the baud rate to 9600 in the BIOS setup so the BIOS and pfSense are both accessible with the same settings. Refer to the manual for your hardware for information on setting its baud rate. 9600 is only the default speed out of the box, you may later increase the serial speed used by pfSense, see the section called “Serial Console Speed”.

## Assign Network Interfaces

After the device is powered on and the boot process has started, a prompt will be shown for VLANs and assigning network interfaces. This step was covered earlier under the section called “Assigning Interfaces” for automatic detection, and later in the section called “Manually Assigning Interfaces” for manually assigning interfaces.

Once the interfaces have been assigned, the system should be ready to configure via the WebGUI.

# Alternate Installation Techniques

This section describes some alternate methods of installation that may be easier for some deployments.

# Installation with drive in a different machine

If it is difficult or impossible to add a CD-ROM drive to the target hardware, and the system cannot boot from USB, another system may be utilized to install pfSense on the target hard drive. The drive may then be moved to the original machine.

When prompted with `Assign Interfaces` during the live CD boot, choose **`n`** for VLANs and type **`exit`** at the assign LAN interface prompt to skip interface assignment. Then proceed through the installation normally. A prompt will appear in the installer for configuring network settings, and this may be skipped as well. After installation, allow the 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 system. After boot, it will prompt for interface assignment and then the rest of the configuration may be performed as usual.

## Boot failure after moving drive to target machine

If the machine used to perform the install assigned the drive with a different device name than the target device, the system will halt booting at a `mountroot>` prompt. This can happen if the install was performed with the drive on the secondary IDE port and in the target hardware it resides on the primary IDE port. In the case of VMware, the USB adapter may be detected as a SCSI device while the target hardware uses IDE.

If this problem is encountered, the system will stop booting and sit at a `mountroot>` prompt, as in this example:

```
Timecounter "TSC" frequency 431646144 Hz quality 800
Timecounters tick every 10.000 msec
Fast IPsec: Initialized Security Association Processing.
ad0: 3906MB <HMS360404D5CF00 DN4OCA2A> at ata0-master UDMA33
Trying to mount root from ufs:/dev/ad2s1a

Manual root filesystem specification:
  <fstype>:<device>  Mount <device> using filesystem <fstype>
                       eg. ufs:da0s1a
```

```
  ?                List valid disk boot devices
  <empty line>     Abort manual input

mountroot> ufs:ad0s1a
Trying to mount root from ufs:ad0s1a


    ___
 ___/ f \
/ p \___/ Sense
\___/   \
    \___/
```

The system is trying to mount the drive by the wrong device name, such as `ad2`. A line just above the `mountroot` prompt should list the real location of the drive, such as `ad0`. To continue the boot process, type in the correct device name. In this case, **ufs:*ad0*s1a**. Just replace *ad0* in that line with the device name of the hard drive as shown above this prompt. Make a note of the proper device name, as it will be needed for the next step.

Now that the system has booted, one more change is needed. The filesystem table in `/etc/fstab` needs to be updated with the proper device. To change this in the WebGUI, browse to Diagnostics → Edit file, and open `/etc/fstab`. Replace each instance of the device name in that file and save your changes. Reboot to verify the change.

For those familiar with command line operations, to change this at the command line choose option **8** once the console menu loads to enter to start a shell. This example uses the **vi** editor. If **vi** is not a desirable choice, **ee** is also available and has on-screen help.

Now enter the command to edit the `fstab` file.

```
# vi /etc/fstab
```

The contents of the file will appear. It will look something like this:

```
# Device            Mountpoint      FStype  Options         Dump    Pass#
/dev/ad2s1a         /               ufs     rw               1      1
```

Make the necessary changes. In this example, the incorrect device is `ad2`, this should be changed to `ad0`:

```
# Device            Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1a         /               ufs     rw              1       1
```

Now save the file and quit the editor. (**Esc**, then **:wq!** if **vi** was used.)

# Full Installation in VMware with USB Redirection

You can use the USB redirection in VMware Player and Workstation to install to a hard drive. Most any USB to IDE or SFF (Small Form Factor) IDE adapter will work for this purpose. The following instructions are specific to VMware Workstation 6.0 and earlier.

- Create a VM with USB redirection.
- Unplug your CF writer from your PC.
- Plug your CF/Microdrive into your CF writer.
- Start the virtual machine, and click inside the VM to give it focus.

- Plug the CF writer into your PC. The VM will pick up the USB device, and the pfSense installer CD will recognize the CF/Microdrive as a hard drive.
- Continue through the installation the same as a normal Full Install.

In VMware Workstation 6.5 and newer, you will see an icon for each USB device on the host along the bottom of the VMware window. Click the device and click **Connect (Disconnect from host)** to use it inside your VM. Refer to the VMware documentation for more information on USB redirection.

## Embedded Installation in VMware with USB Redirection

The embedded image may also be written in VMware using its USB redirection. This is a safer option as it makes it impossible to overwrite disks on the host, limiting potential damage to what is in your virtual machine. To do so, simply attach your CF writer to the VM and perform the installation as you would on the same OS on a physical machine. Refer to the VMware documentation for more information on USB redirection.

## On-the-fly NanoBSD image while booting LiveCD or memstick

If you can boot the LiveCD or memstick, but cannot easily move the target drive to another machine to image the disk, there is another way. You can fetch and write the NanoBSD image in one pass, but doing so is not generally recommended as there is no way to verify the download before writing the image. However, if circumstances force you to use this method, it can be effective.

First, boot the LiveCD or memstick all the way, and configure it such that it has external connectivity. Typically the default config is fine if you're in a DHCP environment. Once you have confirmed external connectivity, go to a shell prompt (**8** at the console menu) and type the following command:

```
# fetch -o - http://files.pfsense.org/mirror/downloads/pfSense-2.1-RELEASE-4g-i
  gzip -dc | dd of=/dev/ad0 obs=64k
```

That command fetched the image, and instead of storing it anywhere, outputs it directly into the decompression and disk imaging process. As with the earlier NanoBSD imaging instructions on FreeBSD (See the section called “Embedded Installation in FreeBSD”), you can also replace the NanoBSD image size, architecture, and if needed, use the VGA NanoBSD images, adjusting the command appropriately.

### Note

This method will only work if you hard drive is ad0, otherwise some manual changes to the filesystem labels would be required.

# 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 taken to resolve them.

## Boot from Live CD Fails

Due to the wide array of hardware combinations in use, it is not uncommon for a CD to boot incorrectly (or not at all). The most common problems and solutions are:

| | |
|---|---|
| Dirty CD-ROM Drive | Clean the drive with a cleaning disc or a can of compressed air, or try another drive. |

| | |
|---|---|
| Bad CD-R Media | Burn another disc and/or burn the disc at a lower speed. Perhaps try another brand of media. |
| BIOS Issues | Update to the most recent BIOS, and disable any unneeded peripherals such as Firewire, Floppy Drives, and Audio. |
| IDE Cable Issues | Try a different IDE cable between the CD-ROM drive and the IDE Controller or Motherboard |
| Boot Loader Issues | There have been cases where specific versions of FreeBSD's CD boot loader will not work on some systems. In this case, see the section above about performing the hard drive installation on a separate PC and then moving it to the target system. |

There are more troubleshooting techniques listed on the pfSense documentation Wiki under Boot Troubleshooting [http://doc.pfsense.org/index.php/Boot_Troubleshooting].

# Boot from hard drive after CD installation fails

After the CD installation completes and the system restarts, there are some conditions which may prevent pfSense from fully booting. The most common reasons are typically BIOS or hard drive controller related. Some of these may be worked around by choosing different options for the boot loader during the installation process, enabling/disabling Packet Mode, or by installing a third party boot loader such as GRUB[1]. Upgrading the BIOS to the latest version available may also help in this case.

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.

As in the previous section, there are more troubleshooting techniques listed in the online documentation under Boot Troubleshooting [http://doc.pfsense.org/index.php/Boot_Troubleshooting].

# Interface link up not detected

If the system complains that interface link up is not detected, 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. You may also want to test or replace the cable in question. After selecting the option, plug the cable back into the interface and ensure it has a link light prior to pressing Enter.

If a network cable is being connected directly between two systems and not to a switch, ensure that a crossover cable [http://en.wikipedia.org/wiki/Ethernet_crossover_cable] is being used. Some newer adapters may support Auto-MDIX [http://en.wikipedia.org/wiki/Auto-MDIX] and will handle this internally, but many older adapters do not. Similarly, if connecting a pfSense system to a switch that does not support Auto-MDIX, use a straight-through patch cable.

If the interface is being properly connected but pfSense still does not detect the link up, the network interfaces being used may not properly detect link for some reason. In this case, manually assigning the interfaces is necessary.

## Manually Assigning Interfaces

If the auto-detection feature didn't work, there is still hope of telling the difference between network cards prior to installation. One way is by MAC address, which should be shown next to the interface names on the assignment screen:

[1]GRUB is a featureful boot loader that supports various operating systems, boot media, and filesystems. Its website is at http://www.gnu.org/software/grub/.

```
le0    08:00:27:26:a4:04
le1    08:00:27:32:ec:2f
```

The MAC address is sometimes printed on a sticker somewhere physically on the network card. MAC addresses also are assigned by manufacturer, and there are several online databases which will let you do a reverse lookup on a MAC address in order to find the company which made the card.[2]

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 `fxp` driver apart from a Realtek card using the `rl` driver by looking at the cards themselves and comparing the names printed upon the circuitry.

Once it is determined which network card will be used for a given role, type it in at the interface assignment screen when prompted. In the above example, `le0` will be WAN and `le1` will be LAN. When prompted first for the WAN address, one would type **`le0`** and press enter. Then when prompted for LAN, type **`le1`**, and press enter. Since there are no optional interfaces, one more press of enter, then **`y`** will complete the assignment. On nearly all tower PCs, the highest PCI slot will be the first NIC, ordered sequentially in top down order. Where you have three Intel `fxp` cards in a system, the top NIC is normally `fxp0`, the one beneath that `fxp1`, and the lowest one `fxp2`. This is dependent on the motherboard, but almost always holds true. If you have an onboard NIC that is the same brand as an add-in NIC, be aware that some systems will list the onboard NIC first, and others will not.

# Hardware Troubleshooting

If you run into problems with the hardware you are attempting to use, the following suggestions will help resolve them in many cases.

## Booting from USB

If the boot stops with a mountroot error 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 and you can proceed with normal installation. On 2.0 and newer this is on the boot menu - option #3 to boot from USB devices.

If you are 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 system contains any hardware that will not be used, remove it. For example, if you have redeployed an old desktop with a sound card, remove the sound card. This normally isn't an issue, but can cause problems and has the potential to reduce performance. If it's removable and you don't need it, take it out of the system.

## Disable PNP OS in your BIOS

This is the most common fix for hardware problems. Many BIOS configuration screens will have 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`**.

[2] http://www.8086.net/tools/mac/, http://www.coffer.com/mac_find/, and http://aruljohn.com/mac.pl, among many others.

## Upgrade your BIOS

The second most common fix for hardware problems is upgrading your BIOS to the latest revision. People seem to have a hard time believing this one, but trust me, just do it. BIOS updates commonly fix bugs in your hardware. It isn't uncommon to hit problems induced by hardware bugs on systems that have stably run Windows for years. I presume either Windows doesn't trigger the bug, or has a work around, as I have personally seen this on multiple occasions. Things that BIOS updates can fix include failing to boot, time keeping problems, and general instability amongst others.

## Reset BIOS settings to factory defaults

Some recycled systems may have an atypical BIOS configuration from its previous use. Most contain an option allowing you to reset all settings to the factory defaults. Try doing this. Also check the section called “Disable PNP OS in your BIOS” again after doing this.

## Disable unused hardware in your BIOS

If your motherboard has any built in components that will not be used, try disabling them. Common examples include the parallel port, onboard modems, audio devices, firewire, possibly USB, and the serial ports unless you plan to use a serial console.

## Other BIOS settings

If your BIOS allows power management configuration, try toggling that option. Look for anything else that seems relevant and try changing some things. If you get to this point, your hardware is probably a lost cause and you should seek alternate hardware. Also check to see if your BIOS has an event log that may list hardware errors such as memory test failures.

## Other Hardware Issues

There could also be some problem with the target hardware, which testing with diagnostic software may reveal. You should test the hard drive with the manufacturer's diagnostic software, and test the memory with a program such as memtest86+. These and more tools are available on the "Ultimate Boot CD [http://www.ultimatebootcd.com/]", 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, some compressed/canned air cleaning of the fans and heat sinks can work wonders.

# Embedded Boot Problems on ALIX Hardware

If an embedded system does not boot properly, connect a serial cable to the device and monitor the boot process for clues on how to proceed. The most common problem will be for users of ALIX hardware. If you are using an ALIX board, you will need to ensure that the latest BIOS available at the time of this writing, 0.99h, is loaded on the board in order to properly boot NanoBSD images from both slices.

An ALIX in need of a BIOS update will typically exhibit the following symptoms on boot:

```
PC Engines ALIX.2 v0.99
640 KB Base Memory
261120 KB Extended Memory

01F0 Master 848A SanDisk SDCFH2-004G
Phys C/H/S 7964/16/63 Log C/H/S 995/128/63

1   FreeBSD
2   FreeBSD
```

```
Boot:   1 ############
```

The number of hash marks (#) will slowly grow over time as the boot attempts to continue. If this behavior is seen, follow the BIOS update procedures from your vendor to at least version 0.99h

In addition to needing BIOS version 0.99h, the BIOS must also be set for CHS mode (Cylinder/Head/Sector mode for addressing data on a disk), as in the following example:

```
PC Engines ALIX.2 v0.99h
640 KB Base Memory
261120 KB Extended Memory

01F0 Master 848A SanDisk SDCFH2-004G
Phys C/H/S 7964/16/63 Log C/H/S 995/128/63

BIOS setup:

*9* 9600 baud (2) 19200 baud (3) 38400 baud (5) 57600 baud (1) 115200 baud
*C* CHS mode (L) LBA mode (W) HDD wait (V) HDD slave (U) UDMA enable
(M) MFGPT workaround
(P) late PCI init
*R* Serial console enable
(E) PXE boot enable
(X) Xmodem upload
(Q) Quit
```

To get to this screen, press **S** while the memory test is displayed over the serial console. Then press **C** to change to CHS mode, then press **Q** to quit.

At this point the ALIX should properly boot from either slice of a NanoBSD image.

# Embedded Boot Problems on Newer Hardware

If you are using NanoBSD on newer hardware or high-end systems, especially with an mSATA or SATA SSD, you might find that the system fails to boot or is unstable. This is due to the embedded image's primary target hardware not supporting some features like DMA or ACPI, whereas more modern or higher quality systems do support those features.

The primary symptom would be repeated disk errors showing on the console, and eventually a potential lockup or extreme performance degradation.

To work around this, you will need to manually re-enable DMA and write caching for the drive. This is done by editing `/boot/loader.conf.local` and adding the following lines:

```
hw.ata.atapi_dma="1"
hw.ata.ata_dma="1"
hw.ata.wc="1"
```

If you are unable to boot the system at all, these can be temporarily set at the loader (1/2 prompt) during boot. Immediately after the loader prompt, press the space bar. If you catch it at the proper moment, you will be presented with a loader prompt. Once you are at the prompt, enter the following commands:

```
set hw.ata.atapi_dma="1"
set hw.ata.ata_dma="1"
set hw.ata.wc="1"
boot
```

Then it should boot normally and allow you to make the required edits to `/boot/loader.conf.local`.

# Recovery Installation

There are two main scenarios for needing to reinstall the system. In the first case, a hard drive or mass storage device may have failed and a fast reinstall with a backup configuration is needed. In the second case, the configuration is still present on the hard drive but some contents of the filesystem may be corrupt. pfSense provides an easy and relatively painless process for recovering quickly from such problems, and if neither of these scenarios applies then there is always the traditional method of restoring a configuration from within the WebGUI.

## Pre-Flight Installer Configuration Recovery

pfSense has, as part of the installation routine, a "Pre-Flight Install" or PFI. PFI will check for an existing configuration on a USB drive, and use it instead of prompting for a new configuration. When installing to a hard drive, the installation program will copy this configuration. When the process is complete, it will restart with the restored configuration file.

First, locate a USB drive that is FAT formatted. If it works in Windows, it is likely already FAT formatted.

Make a directory on the root of this USB drive called `conf`.

Place a configuration file in this folder. If the backup came from the pfSense WebGUI, it is likely named such as this: `config-routerhostname.example.com-20111018142139.xml`. Rename this file to `config.xml`. For more information on making backups, see Chapter 9, *Backup and Recovery*.

The drive should now be ready to use. To double check that the config is in the right place, the file should be in `E:\conf\config.xml` if the USB drive is `E:`. Substitute the appropriate drive letter for the system being used.

Remove the USB drive from the workstation, and then plug it into the pfSense system being restored. Put the Live CD in its CD-ROM drive, and boot the system all the way up to the menu prompt, don't press '**i**' during the boot process to invoke the install early. Once fully booted, it should be noticeable that the system used the configuration from the USB and did not prompt to configure interfaces. The only thing left to do is follow the steps described in the section called “Installing to the Hard Drive” to perform a normal installation to a hard drive.

When the installation is finished, shut down the system, unplug the USB drive, and remove the installation CD. Turn the system back on, and it should boot normally and be fully operational. If any packages were in use, visit the WebGUI and after login they will be automatically reinstalled.

## Installed Configuration Recovery

If portions of the installation on the hard drive are not working (as a result of a failed upgrade or other cause), the configuration may be retained while wiping out the rest of the installed files.

During the install process, before choosing Install pfSense there is a menu choice labeled Rescue config.xml. When this option is chosen, a configuration may be selected from any mass storage media connected to the system. The installation process will load this configuration, and once the reinstallation is complete, the system will be running with the rescued settings.

## WebGUI Recovery

If all else fails, proceed to do a normal installation as described earlier in this chapter then restore the old configuration by visiting Diagnostics → Backup/Restore in the WebGUI once network connectivity has been restored. In the Restore Configuration section of the page, click Browse, find

the configuration backup file. Once located, click Open, and then finally click Restore Configuration. The configuration will be restored and the system will automatically reboot. After rebooting, the full configuration should be present. This process is described in greater detail in the section called “Restoring from Backups”.

# Upgrading an Existing Installation

The supported means of upgrading from one pfSense release to another depend on the platform being used. In most cases, pfSense can be reliably upgraded to any other version while retaining the existing configuration.

By keeping a pfSense system updated with a current supported release, it will never be obsolete. New versions are released periodically that contain new features, updates, bug fixes, and various other changes. In most cases, updating a pfSense installation is very easy. If updating to a new release that is a only a point release (e.g 2.0.2 to 2.0.3), upgrading should be minimally invasive and unlikely to cause any problems. The most common problem 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 1.2.3 to 2.0 should 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 pitfalls one might encounter during such an endeavor. These notes vary from release to release, the most current version can be found on the documentation wiki [http://doc.pfsense.org/index.php/Upgrade_Guide].

## Make a Backup ... and a Backup Plan

First things first, before making any modifications to a pfSense system, it is a good idea to make a backup. In the WebGUI, visit Diagnostics → Backup/Restore. In the Backup Configuration section of the page, ensure that Backup Area is set to **ALL**, then click Download Configuration. Save this file somewhere safe, and it wouldn't hurt to make multiple copies. Those with a pfSense Portal [https://portal.pfsense.org/] subscription should consider using the Auto Config Backup package, and making a manual backup noting the reason as prior to upgrade.

It may also be a good idea to have install media handy for the release currently being run, in case something goes awry and a reinstall is required. Should that happen, have the backup file on hand and refer to the earlier the section called “Recovery Installation”. Also refer to Chapter 9, *Backup and Recovery*.

## Upgrading an Embedded Install

Before version 1.2.3, the only 100% guaranteed reliable way to upgrade embedded was to re-flash the CF and restore a previous configuration backup afterward. That method may still be used, but thanks to the new NanoBSD-based embedded version in use from 1.2.3 forward, reliable upgrades can be performed just like a full install. Continue on into the Full Install upgrade instructions if you are already running pfSense version 1.2.3 or newer.

### Note

If you are updating from an older version of pfSense up to version 1.2.3 or newer, you will still need to re-flash the card with a new NanoBSD-based image. From then on you can update as usual.

## Upgrading a Full Install or NanoBSD install

There are several methods available for updating a Full Installation or NanoBSD installation of pfSense. Either the WebGUI or the console can be used, and either method has a means of supplying a downloaded update file or pulling one automatically from the Internet.

# Upgrading using the WebGUI

There are two options for upgrading using the web interface, with the manual and automatic update. The following sections describe these update methods.

## Manual Firmware Update

In order to perform a manual firmware update, first an update file will need to be downloaded. Browse to http://www.pfsense.org and click the Downloads link. On the Downloads page, click the link for Upgrades. This will lead to the mirror selection page. Pick a mirror geographically close to your location for best performance. Once a mirror has been selected, a directory listing will appear with update files for the current pfSense release. Download the `.tgz` file, (e.g. `pfSense-Full-Update-2.1-RELEASE.tgz`) and the accompanying `.md5` file to verify the download. See the section called "Verifying the integrity of the download" on MD5 for details on how to use an `.md5` file.

To install the update file, visit the pfSense WebGUI. Click System → Firmware. Click Enable Firmware Upload. Click the Browse button next to Firmware Image File. Locate the update file downloaded in the previous step, and click Open. Finally, click the Upgrade Firmware button. The update will take a few minutes to upload and apply, depending on the speed of the connection being used for the update and the speed of the target system. The firewall will reboot automatically when finished.

## Automatic Update

Automatic Update is a new feature that will contact a pfsense.org server and determine if there is a newer released version than the one being run currently. This check is performed when you visit the Automatic Updates page found under System → Firmware, then click the Auto Update tab in the WebGUI. If a new update is available, it will be listed. Click the button to install the update. The update will take a few minutes to download and apply, depending on the speed of the Internet connection being used and the speed of the target system. The firewall will reboot automatically when finished.

By default, the update check only pertains to officially released versions of pfSense, but it is also possible to use this method to track snapshots as well. The update location can be changed by visiting the Updater Settings tab, located immediately to the right of the Auto Update tab. It is safest to use the released versions, as they see the most testing and should be reasonably safe and trouble-free. However, as with any upgrade, you should first visit the pfSense website and read the update notes for that release. When selecting an update URL from the list, be sure to choose the proper architecture (i386 or amd64). If you are unsure of the architecture, check the System Information widget on the dashboard.

As of pfSense 2.0, Automatic Update works on NanoBSD as well as Full Installs.

# 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 pfSense system to be upgraded, start the upgrade process by choosing menu option **13**.

## Update from a URL

If the full URL to a pfSense update file is known, this is a good choice. It will avoid having to first download the update file only to upload it again, and unlike the Automatic Update feature in the WebGUI it also allows a custom update file location to be used.

From the console update menu, choose option **1** for Update from a URL. Enter the full URL to the update file, such as:

```
http://files.pfsense.org/mirror/updates/pfSense-Full-Update-2.1-RELEASE-i386.tgz
```

Confirm that the update should be applied, and then it should be automatically downloaded and installed. After the installation is complete, the router will automatically reboot.

When prompted for a URL to update, you may also enter **auto** which will use the auto-update settings to determine the firmware's URL, rather than manually entering the full path to the image.

### Update from a local file

An update file can be downloaded, as in the manual firmware update above, and then copied to the pfSense system via **scp** or Diagnostics → Command. To install such a file, from the console update menu, choose option **2** for Update From a Local File, and then enter the full path to the file that was uploaded, such as `/root/pfSense-Full-Update-2.1-RELEASE-i386.tgz`. Confirm that the update should be applied, and then it should be automatically installed. After the installation is complete, the router will automatically reboot.

## Upgrading a Live CD Install

On a separate system, download and burn a CD containing the latest release. Ensure that you have moved your configuration to removable media (USB or Floppy) from the console menu (see the section called “Move configuration file to removable device”). Next, restart the pfSense router and boot with the new CD. When pfSense boots on the new CD, the existing storage media containing your configuration will be found and used.

# Filesystem Tweaks

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

## Enabling TRIM Support

The underlying operating system on pfSense 2.1 does support TRIM for flash-based filesystems however we do not yet enable it during the install process. If you believe that TRIM will improve the lifetime of your SSD or other disk that supports TRIM, then it may be enabled by creating an empty file called `/root/TRIM_set` and then rebooting the firewall. The easiest way to do this is via Diagnostics → Command, by running **touch /root/TRIM_set**.

If that file is present at boot time, the firewall will enable trim on the root (`/`) filesystem and the config (`/cf`) slices and then immediately reboot to activate the change, and it will remove the `/root/TRIM_set` file afterward.

Similarly, to remove TRIM support from the drive, create the `/root/TRIM_unset` file.

## 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. 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, you can run **fsck -p** from the shell to determine if the filesystem has any errors. If you find errors, then create the file `/root/force_fsck` and then reboot. That will force pfSense to run a filesystem check during the boot sequence even if the drive isn't considered unclean.

## Speed and Stability Tweaks with sysctl

There are some tunable values that can be changed under System → Advanced on the System Tunables tab.

The **vfs.forcesync** tunable is new for 2.1 and was created because NanoBSD on certain compact flash media was very slow to switch between read-only and read/write. If you find that for some reason your filesystem becomes corrupt during power loss, you can toggle the setting for this value to make it use the slower, but safer, sync method.

You might also encounter situations on NanoBSD where DMA is disabled but it can improve speed. See the section called “Embedded Boot Problems on Newer Hardware”.

# Chapter 5. Configuration

After installation, the pfSense firewall is ready for configuration. The bulk of the configuration is done using the web-based GUI configurator (webConfigurator), or WebGUI for short. There are some tasks that may also be easily performed from the console, whether it be a monitor and keyboard, over a serial port, or via SSH. Some of these may be necessary before you will be able to access the WebGUI, such as if you want to bring up the LAN on an existing LAN network with a different IP address.

# Connecting to the WebGUI

In order to reach the WebGUI, you must connect from another PC. This PC could be directly connected with a crossover cable, or connected to the same switch. By default, the LAN IP 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 PC being used to connect is set to obtain its IP address by DHCP, it should only be a matter of pointing your favorite web browser to https://192.168.1.1. The system uses the IPv6 link-local address of *`fe80::1:1`* by default on LAN, so if you need to reach the WebGUI over IPv6 initially, you may do so by using that address, such as https://[fe80::1:1].

## Note

Some web browsers, notably Firefox, have a bug where they will not accept a self-signed certificate for HTTPS access over IPv6 using the IP in the URL instead of a hostname. In such a case, another browser such as Chrome would work, accessing the firewall over IPv4, or if DNS is functional, access it using the hostname.

If you need to change the LAN IP address or disable DHCP, this may be done from the console by choosing option 2, then enter the new LAN IP, subnet mask, and specify whether or not to enable DHCP. If you choose to enable DHCP, you will also be asked to enter the starting and ending address of the DHCP pool, which could be any range you like inside of the given subnet.

When you disable the DHCP server, you must statically assign an IP address in the pfSense system's LAN subnet on the PC being used for the configuration, 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 PC is connected to the same LAN as the pfSense system, browse to the LAN IP address. Starting with pfSense 2.0, the GUI listens on HTTPS by default, but if you attempt to browse using HTTP, it will redirect you to the HTTPS port instead. If you want to access the GUI directly without the redirect, use https://192.168.1.1.

## Note

Be careful when assigning a new LAN IP address. This IP address **cannot** be in the same subnet as the WAN or any other active interface.

# Setup Wizard

When browsing to the WebGUI, you will first be greeted by a login screen. For the username enter **`admin`** and for the password, enter **`pfsense`**.

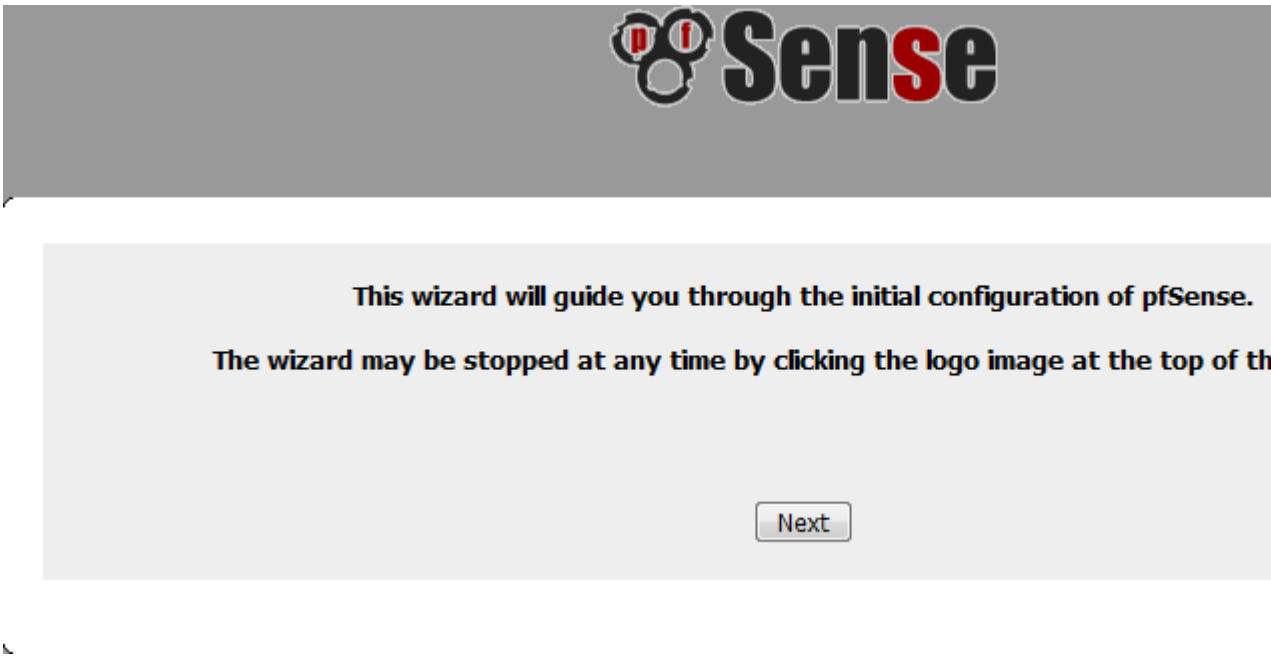
Since this is the first time visiting the WebGUI, the Setup Wizard will begin automatically, and will look like Figure 5.1, “Setup Wizard Starting Screen”. Click Next to start the configuration process.

## Note

Using the setup wizard is optional. If you need to create a more complex configuration or if the default values are acceptable, you may simply click the logo at the top of the

wizard to get back to the firewall configuration. Once out of the wizard, you may then make any needed adjustments manually from there.

**Figure 5.1. Setup Wizard Starting Screen**

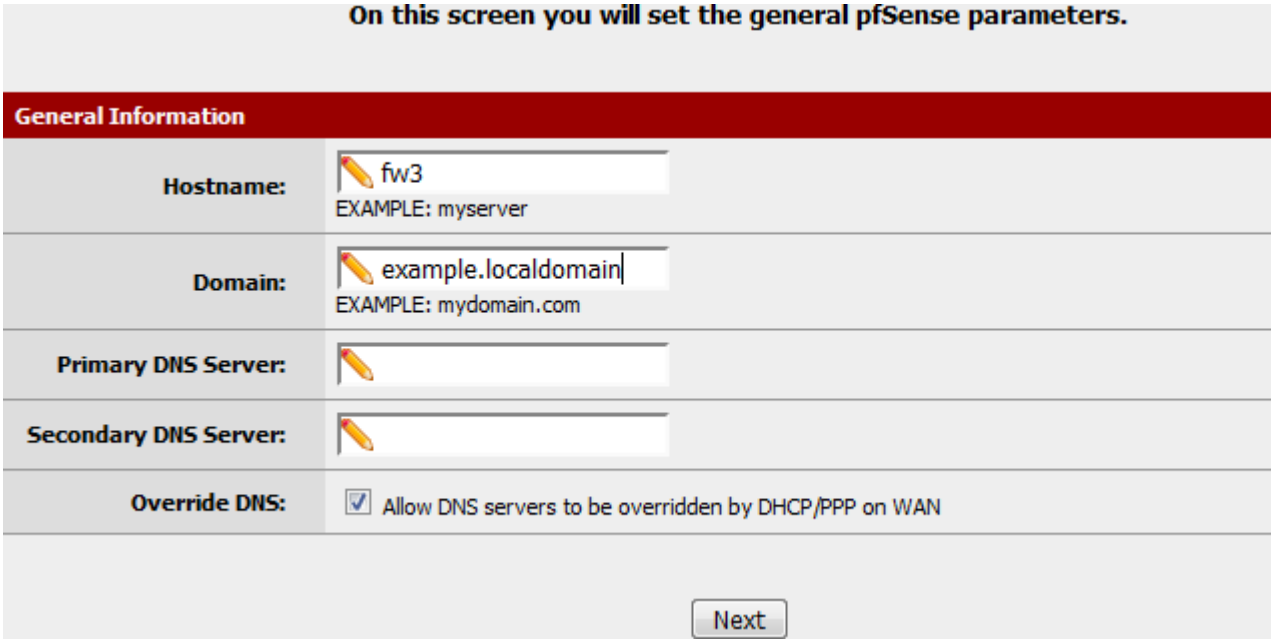



# General Information Screen

The next screen (Figure 5.2, “General Information Screen”) will ask for the name of this pfSense router, and the domain in which it resides. The Hostname can be anything you like, but must start with a letter, and then it may contain only letters, numbers, or a hyphen. After the hostname, enter a Domain, e.g. **`example.com`**. If you do not have a domain, you can use **`<something>.localdomain`**, where *`<something>`* is anything you want: a company name, your last name, nickname, etc. The hostname and domain name are combined to make up the fully qualified domain name of your firewall.

The Primary DNS Server and Secondary DNS Server may be filled in, if known. If you are using a dynamic WAN type such as DHCP, PPTP or PPPoE connections, these will usually be automatically assigned by your ISP and can be left blank. These WAN types are explained in more detail later in the Setup Wizard. Click Next when finished.

**Figure 5.2. General Information Screen**

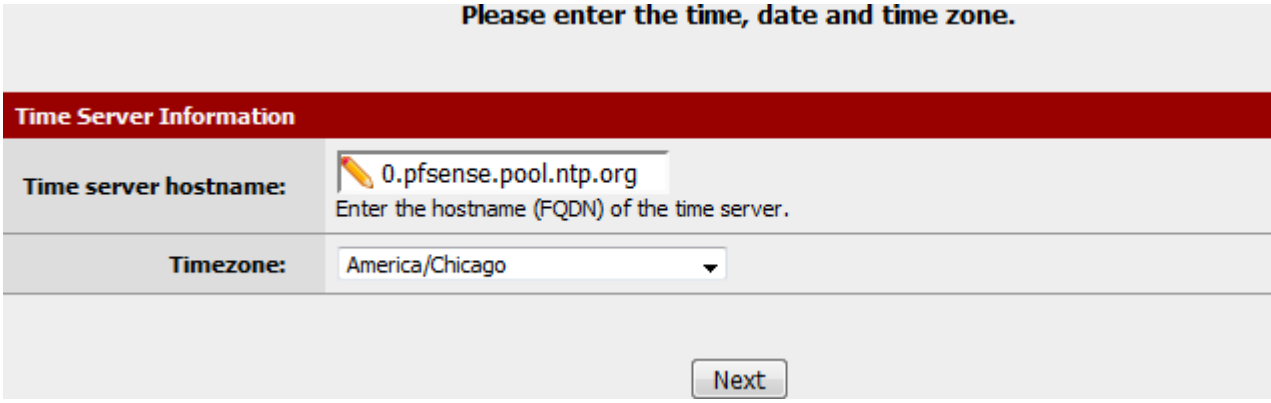



# NTP and Time Zone Configuration

The next screen (Figure 5.3, “NTP and Time Zone Setup Screen”) has a place for a Network Time Protocol (NTP) server, and the time zone in which this server resides. Unless you have a specific preference for an NTP server such as one inside your LAN, it is best to leave the Time server hostname at the default **`0.pfsense.pool.ntp.org`**, which will pick a random server from a pool of known-good NTP hosts. If multiple time servers are desired, they may be added in the same box, separating each server by a space. For example, if you want four NTP servers, enter **`0.pfsense.pool.ntp.org 1.pfsense.pool.ntp.org 2.pfsense.pool.ntp.org 3.pfsense.pool.ntp.org`**. The numbering is specific to how **`*.pool.ntp.org`** operates and ensures each address is drawn from a unique pool of IPs so the same server does not get used twice.

For the Timezone selection, choose a geographically named zone which best matches the pfSense system's location. When finished, click Next to continue.

**Figure 5.3. NTP and Time Zone Setup Screen**




# WAN Configuration

These next few paragraphs and their associated images will help guide you through setting up the WAN interface on the pfSense system. Since this is the side facing your ISP or upstream router, there

are configuration choices to support several common ISP connection types. The first choice is for the WAN Type (Figure 5.4, “WAN Configuration”). This should match whatever your ISP supports, or whatever your previous router was configured to use. Possible choices are Static, DHCP, PPPoE, and PPTP. The default choice is DHCP since it is very common, and in most cases will allow a router to "Just Work" without additional configuration. If you are not sure which WAN type to use, or which fields to configure, you will need to obtain this information from your ISP. If your WAN type is not available in the wizard, or you need more information about the WAN types found here, you will find a lot more detailed information in Chapter 6, *Interface Types and Configuration*.

## Note

If you have a wireless interface for your WAN interface, some additional options may appear which are not covered during this walkthrough of the standard Setup Wizard. You may refer to Chapter 23, *Wireless*, which has a section on Wireless WAN for additional information. You may need to skip the WAN setup for now, and then perform the wireless configuration afterward.

### Figure 5.4. WAN Configuration


The MAC Address field in the next section (Figure 5.5, “General WAN Configuration”) is useful for replacing an existing router with minimal complications. Some ISPs, mainly those run by Cable providers, will not work properly if a new MAC address is encountered. Some require power cycling the modem, others require registering the new address with them over the phone. 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.

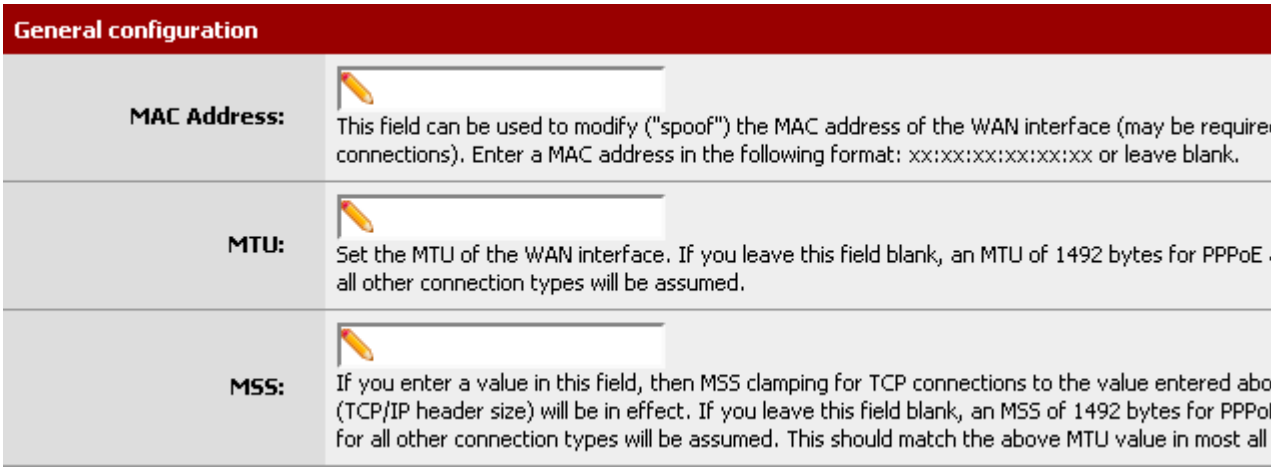
## Note

If you ever intend on using this firewall as part of a high availability cluster (See Chapter 25, *Firewall Redundancy / High Availability*), do not spoof the MAC address.

The Maximum Transmission Unit (MTU) size field seen in Figure 5.5, “General WAN Configuration” can typically be left blank, but can be changed if desired. Some situations may call for a lower MTU to ensure packets are sized appropriately for your Internet connection. In most cases, the default assumed values for the WAN connection type will work properly.

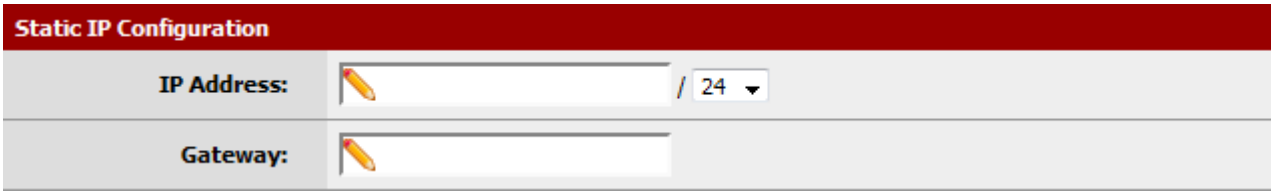
The Maximum Segment Size (MSS) field seen in Figure 5.5, “General WAN Configuration” can typically be left blank, but can be changed if needed. This field enables MSS clamping, which ensures TCP packet sizes remain adequately small for a particular Internet connection. This is usually only set on PPPoE type WANs, where the standard default MSS clamping is inadequate for a particular Internet provider.

### Figure 5.5. General WAN Configuration



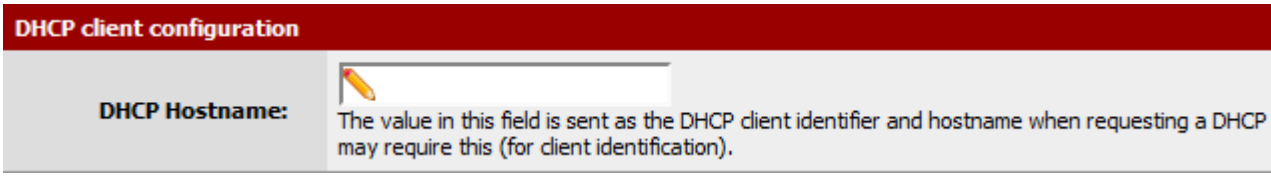

If the "Static" choice for the WAN type is chosen, the IP address, CIDR Subnet mask, and Gateway must all be filled in (Figure 5.6, “Static IP Settings”). This information should be obtained from your ISP or whoever controls the network on the WAN side of your pfSense router. The IP Address and Gateway must both reside in the same Subnet.

### Figure 5.6. Static IP Settings




Some ISPs require a certain DHCP hostname (Figure 5.7, “DHCP Hostname Setting”) to be sent along with the DHCP request to obtain a WAN IP. If you are unsure of what to put in this field, try leaving it blank unless directed otherwise by your ISP.

### Figure 5.7. DHCP Hostname Setting

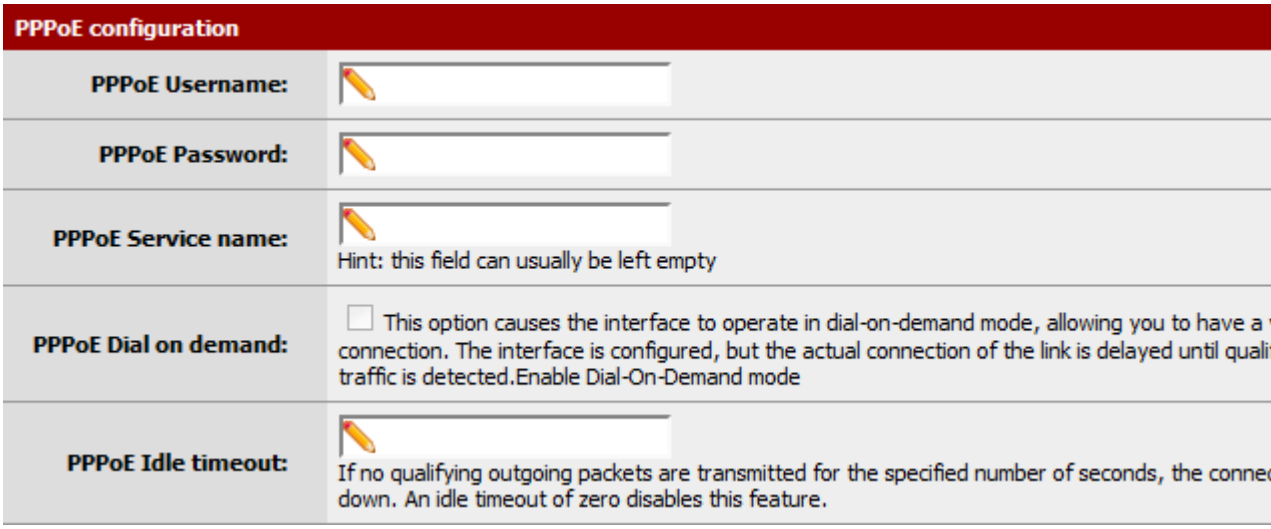



When using the PPPoE (Point-to-Point Protocol over Ethernet) WAN type (Figure 5.8, “PPPoE Configuration”), you must at least fill in the fields for PPPoE Username and PPPoE Password. These will be provided by your ISP, and typically are in the form of an e-mail address, such as **`mycompany@ispexample.com`**. The PPPoE Service name may be required by some ISPs, but is often left blank. If you are in doubt, leave it blank or contact your ISP and ask if it is necessary.

PPPoE Dial on demand will cause 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 you plan on running any services behind the pfSense box, **do not** check this, as you will want to maintain an online connection as much as possible in that case. Also note that this choice will not drop an existing connection.

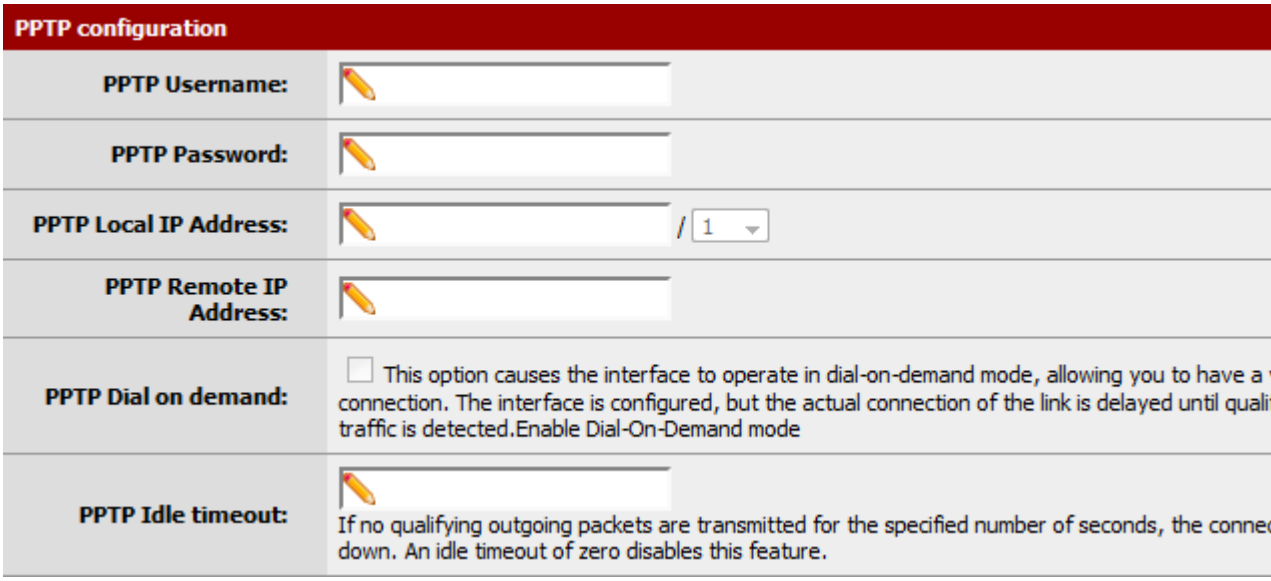
The PPPoE Idle timeout specifies how much time pfSense will let the PPPoE connection go without transmitting data before disconnecting. This is really only useful when coupled with Dial on demand, and is typically left blank (disabled).

### Figure 5.8. PPPoE Configuration



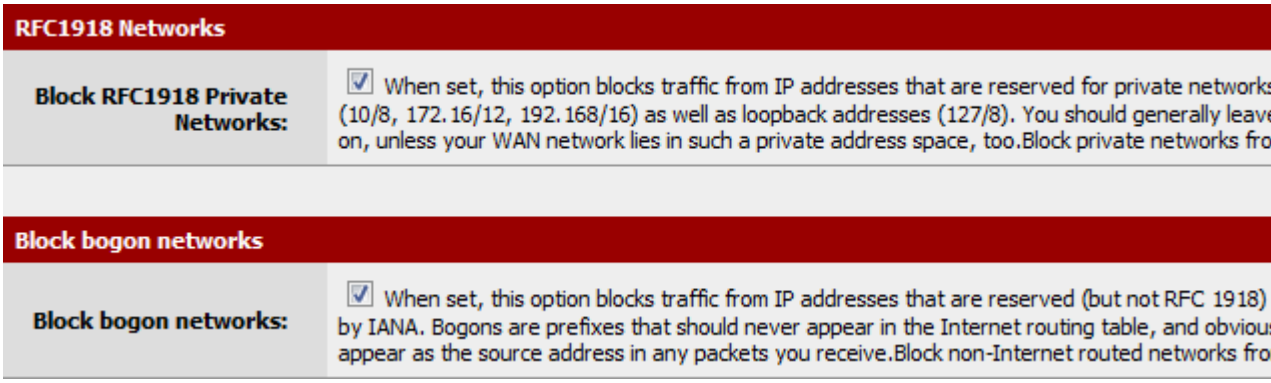

The PPTP (Point-to-Point Tunneling Protocol) WAN type (Figure 5.9, “PPTP WAN Configuration”) is the source of some confusion. This option is for ISPs that require a PPTP login, and **not** for connecting to a remote PPTP VPN. These settings, much like the PPPoE settings, will be provided by your ISP. Unlike PPPoE, however, with a PPTP WAN you must also specify a Local IP address, CIDR subnet mask, and Remote IP Address to establish the connection.

### Figure 5.9. PPTP WAN Configuration




These last two options, seen in Figure 5.10, “Built-in Ingress Filtering Options”, are useful for preventing invalid traffic from entering your network, also known as "Ingress Filtering". Enabling Block RFC 1918 Private Networks will block registered private networks such as 192.168.x.x and 10.x.x.x from making connections to your WAN address. A full list of these networks is in the section called “Private IP Addresses”. The Block bogon networks option will stop traffic from coming in that 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 the section called “Block Bogon Networks”. Click Next to move on when finished.

**Figure 5.10. Built-in Ingress Filtering Options**

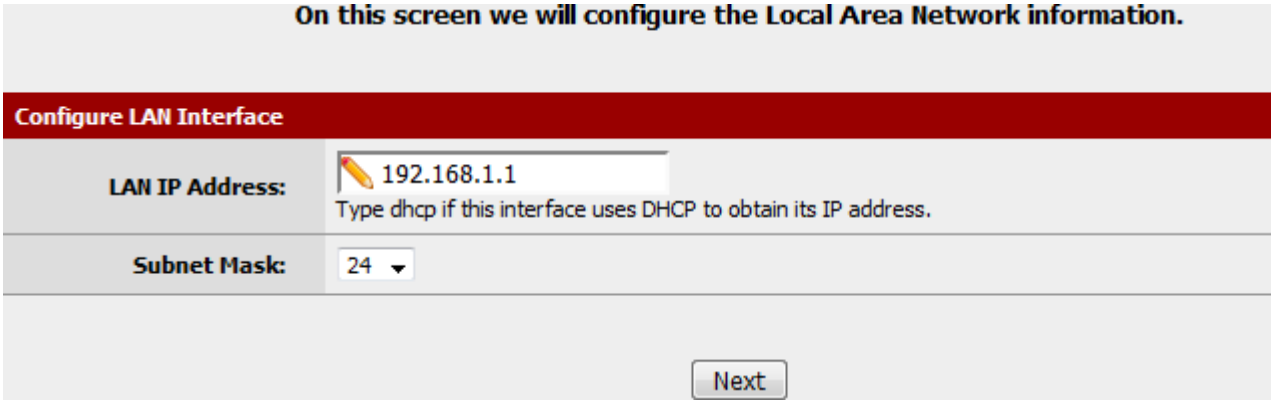



# LAN Interface Configuration

Here you are given an opportunity to change the LAN IP Address and Subnet Mask (Figure 5.11, “LAN Configuration”). If you don't ever plan on connecting your network to any other network via VPN, the default is fine. If you want to be able to connect into your network using VPN from remote locations, you should choose a private IP address range much more obscure than the very common 192.168.1.0/24. Space within the 172.16.0.0/12 RFC 1918 private address block seems to be the least frequently used, so choose something between 172.16.x.x and 172.31.x.x for least likelihood of having VPN connectivity difficulties. If your LAN is 192.168.1.x and you are at a wireless hotspot using 192.168.1.x (very common), you won't be able to communicate across the VPN — 192.168.1.x is the local network, not your network over VPN.

If the LAN IP needs to be changed, enter it here along with a new subnet mask. Be aware that if you change these settings, you will also need to adjust your PC's IP address, release/renew its DHCP lease, or perform a "Repair" or "Diagnose" on the network interface when finished with the setup wizard.

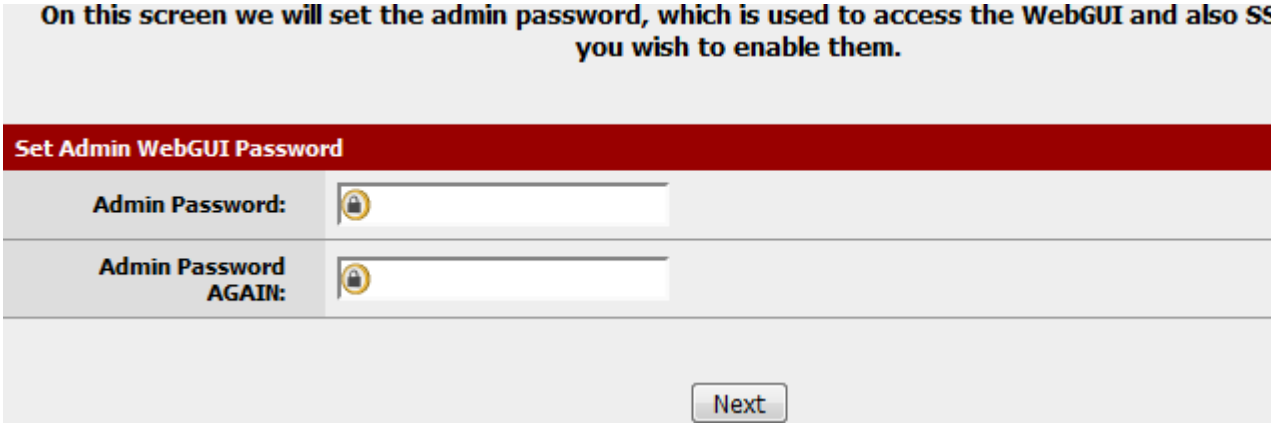
**Figure 5.11. LAN Configuration**




# Set admin password

Next you must change the administrative password for the WebGUI as shown in Figure 5.12, “Change Administrative Password”. This password should be something strong and secure, but no restrictions are automatically enforced. Enter the password twice to be sure that has been entered correctly, then click Next.

**Figure 5.12. Change Administrative Password**




# Completing the Setup Wizard

That's the end of the setup wizard, click Reload (Figure 5.13, “Reload pfSense WebGUI”) and the WebGUI will reload. If you changed the LAN IP, adjust your PC's IP address accordingly. You will also be prompted for the new password. The username is still **`admin`**.

**Figure 5.13. Reload pfSense WebGUI**


At this point you should have basic connectivity to the Internet, or the network on the WAN side. Clients on the LAN side should be able to reach sites through the pfSense router. If at any time you need to repeat this initial configuration, you may do so by going to System → Setup Wizard from within the WebGUI.

# Interface Configuration

As you have seen, some interface configuration can be performed at the console and in the setup wizard to start things out, but changes may also be made after the initial setup by visiting the appropriate places under the Interfaces menu. A few basics are covered here, the details can be found in Chapter 6, *Interface Types and Configuration*.

# Assign interfaces

If additional interfaces are added post-setup, then they may be assigned roles by visiting Interfaces → (assign). There are many tabs here for assigning and creating various types of interfaces. The two most commonly used tabs are Interface assignments and VLANs. (VLAN configuration is covered later in Chapter 14, *Virtual LANs (VLANs)*.) The Interface assignments tab shows a list of all currently assigned interfaces: WAN , LAN, and any OPTx that are configured. Next to each interface is a drop-down list of all network interfaces/ports found on the system, including real hardware interfaces as well as VLAN interfaces and other virtual interface types. The MAC address or VLAN tag will show along side the interface name to aid in identification. The other tabs (Interface Groups, Wireless, QinQs, PPPs, GRE, GIF, Bridges, and LAGG), much like the VLAN tab, are there to create additional interfaces which can then be assigned. All of these interface types are covered in Chapter 6, *Interface Types and Configuration*.

You may change the currently assigned interfaces by picking a new network port, or add an additional OPTx interface by clicking . This will add another line, with a new OPT interface, numbered higher than any existing OPT interface, or if there are none, OPT1. By default, it will automatically choose the next available interface that is unassigned. For example, if the target system has `fxp0`, `fxp1`, and `fxp2`, and you have WAN set to `fxp0`, and LAN set for `fxp1`, choosing to add another interface will automatically assume OPT1 will be `fxp2`. If you have additional interfaces and this is not the intended setting, it may be altered. If any changes are made, be sure to click Save.

# 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. 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. If Static IPv4 is chosen, an IPv4 Address, subnet mask, and gateway may be set. If the other options are chosen, then type-specific fields appear to configure each type. The IPv6 Configuration Type may also be changed to either Static IPv6, DHCP6, SLAAC, 6rd Tunnel, 6to4 Tunnel, or none to leave IPv6 unconfigured on that interface. When choosing Static IPv6, you may set an IPv6 address, prefix length, and gateway. If this a wireless interface, many more options will appear to configure the wireless portion of the interface, as well as the standard configuration items.

In previous versions of pfSense (1.2.x and before), the WAN, LAN, and OPT interface configurations differed. Starting with pfSense 2.0, every interface is configured identically. 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 now be renamed as well, including WAN and LAN, to a custom name of your preference. Furthermore, every interface can be enabled and disabled as desired, so long as one interface remains enabled.

For more information on interface options, see the section called “Interface Configuration”.

### Note

Selecting a Gateway from the drop-down list, or adding a new gateway and selecting it, will make pfSense 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. You may still use gateways on those interfaces for the purpose of static routes without selecting a Gateway here on the interfaces screen.

# Managing Lists in the GUI

The pfSense GUI has a common set of icons which are used for managing lists and collections of objects throughout the system. Not every icon is used in every page, but their meanings are typically

consistent based on the context in which they are seen. Examples of such lists include firewall rules, NAT rules, IPsec or OpenVPN instances, and certificates.

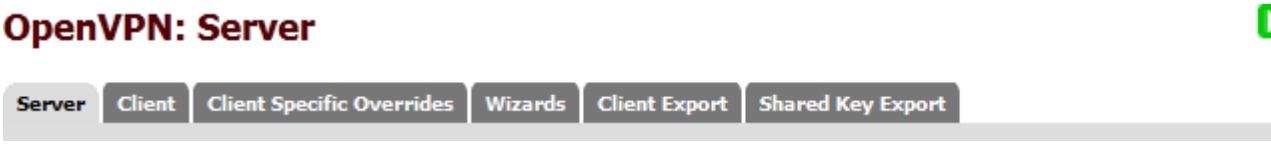
| | |
|---|---|
| (+) icon | Add an item to a list. At the top of a list of ordered items, the + will add an item to the beginning of the list; at the bottom, the + will add to the end. On unordered lists the top and bottom + will simply add a new item. |
| (e) icon | Edit an entry. |
| (x) icon | Delete an entry. |
| (up) icon | Move an item up one row in an ordered list. Also used to indicate an upload or import function in some places. |
| (down) icon | Move an item down one row in an ordered list. Also used to indicate a download or export function in some places. |
| (left) icon | Used for moving items. When working with an ordered list, selected items will be moved to the row above this arrow when clicked, as indicated in the GUI with a dark horizontal bar indicating the position for the new item when the mouse is hovering over this icon. When working with side-by-side lists, this button will move an item from the right list to the left list. |
| (right) icon | Used for moving items. When working with side-by-side lists, this button will move an item from the left list to the right list. |
| (question mark) icon | Indicates a link to sources for help or additional information. |

If you are unsure what action an icon will perform, hover over the icon with your mouse pointer and a tooltip will display a short description of what will be done when the icon has been clicked.

# Quickly Navigate the GUI with Shortcuts

Many areas of the GUI have a shortcut bar present, as seen in Figure 5.14, “Shortcut Bar Example”. These shortcuts can be used to navigate to related pages within the section being viewed.

**Figure 5.14. Shortcut Bar Example**




For example, in Figure 5.14, “Shortcut Bar Example”, the shortcuts would have the following effects:

| | |
|---|---|
| Service Status Indicator (Running) | This icon is present if the service on this page is currently running. |
| Service Status Indicator (Stopped) | This icon is present if the service on this page is currently stopped. |
| Start Service | If the service is stopped, this icon allows you to start the service. |
| Restart Service | If the service is running, this icon allows you to stop and restart the service. |
| Stop Service | If the service is running, this icon allows you to stop the service. |

| | |
|---|---|
| Main Page | When this icon appears, it will take you back to this section's main page. Typically the page containing configuration options for this section. |
| Status Page Link | This icon is a link to this section's status page, 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 now has shortcut controls as well, to aid in navigation to pages related to a given service, as shown in Figure 5.15, “Shortcuts on Service Status”. The icons have the same meaning as in the above section.

**Figure 5.15. Shortcuts on Service Status**


These shortcuts cut down on the amount of hunting one has to do in order to locate pages related to the one currently being viewed. One can 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.

# General Configuration Options

Some general system options are found under System → General Setup; most of them will look familiar from the Setup Wizard.

The Hostname and Domain, DNS Servers, and the Time zone and NTP Time server can be changed if desired, as covered in the Setup Wizard.

On this screen, you may add up to four DNS servers, and in addition to specifying their IPs, you may also select which gateway to use to reach each server. This is especially useful in a Multi-WAN scenario where you ideally have at least one DNS server per WAN. More information on DNS for Multi-WAN can be found in the section called “DNS Servers and Static Routes”.

Along with the ability to change the DNS Servers, there is another option: Allow DNS server list to be overridden by DHCP/PPP on WAN. This does essentially what it says; if checked, pfSense will use the DNS servers that are assigned dynamically by DHCP or PPP. They will be utilized by the system itself and as the upstream DNS servers for the DNS forwarder. These servers will not be passed on to the DHCP clients behind the pfSense system, however.

pfSense 2.0 and newer will also consult the firewall's own DNS Forwarder by default for DNS. This gives the benefit of being able to resolve local addresses from DHCP registrations and DNS overrides, as well as consulting all possible DNS servers at once. For some this behavior is undesirable, and it may be disabled by checking Do not use the DNS Forwarder as a DNS server for the firewall.

In versions of pfSense before 2.0, the admin username and password could be set on this screen. Now those options are managed from within the User Manager (Chapter 7, *User Management and Authentication*). The option to change the WebGUI port and WebGUI Protocol have moved to under System → Advanced with the remainder of the GUI control options.

New in pfSense 2.1 is the ability to select a Language for the GUI. The default is **`English`** and the only other currently available language is **`Portuguese (Brazil)`**.

Lastly, a Theme may also be chosen. Several are included in the base system, and they only make cosmetic — not functional — changes to the look and feel of the WebGUI. The default theme was changed in pfSense 2.0 to be `pfsense_ng`. Several other themes were added in that release as well, contributed by members of the community.

# Advanced Configuration Options

Under System → Advanced you will find a lot of options that are of a more advanced nature. None of these options should need adjustment for a basic routing/NAT setup, but you may find that some of the changes governed by these options will help in customizing your configuration in beneficial ways.

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

In pfSense 1.2.3 and before, these options were all found on a single page. Due to the number of options and size of the page, this has been split into multiple tabs, and some options have been converged here from other areas as well.

# 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 would be to use HTTPS so that the WebGUI traffic is encrypted, especially if the firewall will be managed remotely.

### SSL Certificate

If HTTPS is chosen, you must also select a certificate from the SSL Certificate drop-down list. Out of the box, you should have at least one choice here, webConfigurator default. The default certificate is self-signed. That is not an ideal situation, but is better than no encryption at all. This option allows you to use an existing certificate to further enhance security and protect against man-in-the-middle attacks. If you have an existing SSL certificate and key, you may import them using the Certificate Manager, and then select the certificate here. You may also make additional certificates by using the Certificate Manager if needed.

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 your 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 that also allows a permanent bypass.

### TCP Port

Moving the WebGUI to an alternate port is preferred by some for security by obscurity reasons, though such practices should not be considered any security benefit. It can free up the standard web ports for use with port forwards or other services such as a Squid proxy. By default the WebGUI uses HTTPS on port 443 with a redirect from port 80 for the best compatibility and ease of initial configuration. If you would like to change the port, enter a new port number into the TCP Port field.

### Max Processes

If you find that you have multiple people viewing the GUI at the same time and some pages are taking too long to load, or failing to load, then you may need to increase the Max Processes value. By default it is set to 2, so it runs two web server processes.

## WebGUI Redirect

By default, for ease of access and compatibility, the firewall runs a redirect on port 80 so that if you attempt to access the router with HTTP, it will accept the request and then redirect the session to HTTPS on port 443. This redirect can be disabled by checking Disable webConfigurator redirect rule if something else needs to bind to port 80 or there is another reason it would be unwanted.

## WebGUI Login Autocomplete

For convenience, the login form allows autocomplete so your browser can save the login credentials. In some environments, such as those needing to comply with specific security guidelines, this behavior is not acceptable. As such, it can be disabled by checking Disable webConfigurator login autocomplete. Be aware, however, that not all browsers respect this option. Specifically, Opera will still offer to save passwords even when the form specifies that it should not be allowed. This only controls autocomplete on the login form. Even if autocomplete is enabled for login, the completion of forms elsewhere in the GUI remains disabled to prevent the browser from automatically filling your credentials into unrelated fields.

## WebGUI login messages

Successful logins will result in a message being printed to the console, and on some hardware this will also cause a "beep" to be heard from the device when such a console message occurs. If you do not wish to see this message (or hear the resulting beep), check the Disable logging of webConfigurator successful logins box.

## Anti-lockout

By default, access to the WebGUI and SSH on the LAN interface is always permitted, regardless of the user-defined filter rules. Enabling this feature will allow more fine-grained control over which LAN IP addresses may access the WebGUI, but be sure you have a filter rule in place to allow access before enabling this option!

### Note

Resetting the LAN IP from the system console will also reset this option. If you find yourself locked out after enabling this, choose the console menu option to set the LAN IP, and enter in the exact same IP address and accompanying information.

In pfSense 1.2.x this rule allowed all traffic to reach the firewall's LAN IP address. In pfSense 2.x, this behavior is restricted to only allowing access to the GUI port and the SSH port. In pfSense 2.0 and later, if you only have one interface enabled, the anti-lockout rule will become active on that interface.

## DNS Rebind Check

The remaining options in this section control some of the new security measures put in place to protect the GUI against various means of browser-based attacks. The first of these is DNS Rebinding Checks. When set, this blocks private IP responses from your configured DNS servers. Check this box to disable this protection if it interferes with webConfigurator access or name resolution in your environment. More detail on DNS rebinding attacks may be found on Wikipedia [http://en.wikipedia.org/wiki/DNS_rebinding]. To disable this behavior, check Disable DNS Rebinding Checks. The most common cause for disabling this would be when your firewall is set to use an internal DNS server which will return private (RFC1918) answers for hostnames. If you access the firewall by IP address, these checks are not enforced because the attack is only relevant when using a hostname.

## Browser HTTP_REFERER enforcement

The GUI also 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 you did not intend for that to happen. This

also breaks some desirable behavior, such as having a page that links to your various firewall devices. If you wish to disable this behavior, you may check Disable HTTP_REFERER enforcement check.

### Alternate Hostnames

If you wish to keep the above options enabled, but control the behavior slightly, you may fill in alternate hostnames in the Alternate Hostnames for DNS Rebinding and HTTP_REFERER Checks box. By default the system will allow access to the hostname configured on the firewall and all IPs configured on the system.

### Man-In-The-Middle Attack/Warning

If you attempt to access the firewall by an IP address that is not configured on the system, such as a port forward from another firewall, a message will be printed that you may be the result of a Man-In-The-Middle (MITM) attack. If you configured such a forwarding yourself, the message may be safely ignored. If you did not configure the forward, then you should take great care before logging in to ensure your login credentials are not being routed through a system you do not control or trust. Access is not disabled in this case, only a warning, so there is no option to disable this behavior.

## Secure Shell (SSH)

The Secure Shell (SSH) server may be enabled which will allow remote console and file management. You may connect with any standard SSH client, such as the OpenSSH command line **ssh** client, PuTTY, SecureCRT, or iTerm. Either the WebGUI username (such as admin) or the root account may be used, and both accept the WebGUI password for login. If you have created other users in the User Manager and they also have the User - System - Shell account access permission, then they are allowed to login over ssh as well.

File transfers to and from the pfSense system are also possible by using a Secure Copy (SCP) client such as OpenSSH's command line **scp**, FileZilla, WinSCP or Fugu. To use SCP, you must connect as the root user, not admin. If you have created a user that has the User - System - Copy files permission, or all access, then they may also utilize SCP.

### Enable Secure Shell

To enable SSH, check the box next to Enable Secure Shell. After you save with this setting checked, the system will generate SSH keys (if they are not already present) and start the SSH daemon.

### Authentication Method

You can also set SSH to only allow key-based logins and not a password. Switching to only key-based login is a much more secure practice, though it does take a little more preparation to configure. To do this, check Disable Password login for Secure Shell (RSA KEY only). You add keys for key-based login by editing your users in the User Manager (Chapter 7, *User Management and Authentication*) and pasting the allowed public keys into the Authorized Keys text field for their account. In pfSense 1.2.x and earlier, there was a box here to paste the key since there was only one account (`admin`).

### SSH Port

It is also more secure to move the SSH server to an alternate port. As with moving the WebGUI to an alternate port, it provides a small security improvement, and frees up the port if you want to forward it to an internal system. To change the port, type the new port into the SSH Port box.

### Best Practices for SSH

Should you find yourself in a situation that requires leaving SSH access unrestricted by firewall rules, which can be dangerous, it is highly recommended that in this situation you both move the SSH service to an alternate random port, and switch to key-based authentication. Moving to an alternate port will prevent log noise from brute-force SSH login attempts and casual scans. It can still be found with a port scan, so switching to key-based authentication should 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 will result in locking out the IP address trying to authenticate, but that alone should not be considered sufficient protection.

## Serial Console

If this pfSense system will be running "headless" (without keyboard, video, mouse attached) it may be desirable to enable this option, which will redirect the console input/output to the serial port. This will not disable the onboard keyboard, video, and mouse but will allow you to attach a null modem cable to the serial port and manage it directly from another PC or serial device. After making any changes, be sure to click Save when finished.

With both the serial console enabled and a monitor attached, the serial console is preferred, so it will receive the boot log messages from pfSense. Some other OS kernel messages will show up on all console connections, and both consoles will have a usable menu.

For more information on connecting to a serial console, see the section called “Connect a Serial Cable” and the section called “Start a Serial Client”.

### Serial Console Speed

New in pfSense 2.1 is the ability to change the serial console speed. Previously, this was always locked to 9600bps. Now it can be set higher by selecting a faster speed from the drop-down menu. The fastest possible speed is 115200bps.

## Console Menu

Normally, the console menu is always showing on the system console, and will be available as long as you have physical access to the serial or video console. In some situations this is not desirable, so this option will allow the console to be password protected. You may login with the same username and password used for the WebGUI. After setting this option, you must reboot the pfSense system before it will take effect.

**Note**

While this will stop accidental keypresses, and keep out casual users, this is by no means a perfect security method. A knowledgeable person with physical access could still reset the passwords (see the section called “Forgotten Password with a Locked Console”). You should consider other physical security methods if that is a requirement of your installation.

# Firewall/NAT Tab

## Firewall Advanced

### IP Do-Not-Fragment compatibility

This 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. This will cause the filter to not drop such packets but instead clear the don't fragment bit. The filter will also randomize the IP identification field of outgoing packets with this option on, 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

There are a few choices here that control how the firewall expires states:

| | |
|---|---|
| Normal | The standard optimization algorithm. |
| 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. |

## Disable Firewall

If you choose to Disable all packet filtering, it will turn your pfSense system into a routing-only platform. As a consequence, NAT will also be disabled. If you only wish to disable NAT, do not use this option, rather see the section called “Disabling Outbound NAT” for more information on controlling outbound NAT behavior.

## Disable Firewall Scrub

Disables the PF scrubbing option which can sometimes interfere with NFS and PPTP traffic. By default, pfSense uses the random-id scrub option which randomizes the IP identification field of a packet for added security, and the fragment reassemble option which will reassemble fragmented packets before sending them on to their destination. More information on the Scrub feature can be found on the OpenBSD PF Scrub Documentation [http://www.openbsd.org/faq/pf/scrub.html].

### Note

Disabling scrub will also disable other features that rely on scrub to function, such as DF bit clearing, ID randomization, and MSS clamping.

## Firewall Maximum States

Sets the maximum number of connections to hold in the firewall state table. The default is dynamic, and sized at 10% of the amount of RAM in the system. This default value should be sufficient for most installations, but can be adjusted higher or lower depending on the load and memory available. Each state consumes about 1 KB of RAM, or roughly 1 MB of RAM for every 1000 states, so ensure you have adequate free RAM before increasing this. Firewall states are discussed further in the section called “Stateful Filtering”.

## Firewall Maximum Tables

Maximum number of tables that can exist on the system for items such as aliases. Note that this is the number of tables/aliases themselves, not the count of items inside aliases (that's the next option). By default this is 3,000 entries, which due to various ways the system operates effectively limits you to around 1500 aliases. This can be increased as needed with little to no impact on other resources, but having that many aliases would also be difficult to manage in the GUI.

## Firewall Maximum Table Entries

Maximum number of entries that can exist inside of tables for systems such as aliases, sshlockout, snort, etc. By default this is 200,000 entries. If you use features such as URL Table aliases to load

large blocks of address space into aliases, then you may need to increase this value. Each table entry will consume some amount of RAM, so be careful not to set it arbitrarily high.

### Static Route Filtering

The Bypass firewall rules for traffic on the same interface option only applies if you have defined one or more static routes. If it is enabled, traffic that enters and leaves through the same interface will not be checked by the firewall. This may be required in some 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 the section called “Bypass Firewall Rules for Traffic on Same Interface” for a more in-depth discussion on that topic.

### Disable Auto-added VPN rules

This disables automatically added rules for IPsec and PPTP. Normally, when you enable one of these VPNs, rules are automatically added to the appropriate interface which will allow traffic in to those ports. By disabling these automatic rules, you can have more control over which addresses are allowed to connect to the VPN. Further information on these rules can be found at the section called “VPNs and Firewall Rules”.

### Disable Reply-To

With Multi-WAN you generally want to ensure traffic leaves the same interface it arrives on, hence **reply-to** is added automatically by default to ensure this association for return traffic. When using bridging, you must disable this behavior if the WAN gateway IP is different from the gateway IP of the hosts behind the bridged interface. Another use case involves static routing to other systems in a larger WAN subnet where this option would help ensure that replies went back to the proper system instead of being routed back to the gateway.

### Disable Negate rules

With Multi-WAN you generally want to ensure traffic reaches directly connected networks and VPN networks when using policy routing. pfSense will insert some rules to pass this local and VPN traffic without a gateway specified, to maintain connectivity. You can disable this for special purposes but if you do so, you must manually create rules for these networks without a gateway set so they may be reached without policy routing.

### Aliases Hostnames Resolve Interval

This option controls how often hostnames in aliases are resolved and updated. By default this is *300* seconds (5 minutes). If you do not have very many hostnames, and you need more frequent updates, this can be reduced.

## 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 the section called “Block bogon networks”.

## Network Address Translation

### NAT Reflection for Port Forwards

The NAT Reflection mode for port forwards option controls how NAT reflection rules are made. These NAT redirect rules allow you to access port forwards on your public IP addresses from within your internal networks. These rules are disabled by default, so NAT Reflection rules are not created unless you change this setting. Refer to the section called “NAT Reflection” for a discussion on the merits of NAT Reflection when compared to other techniques such as Split DNS.

The NAT + proxy mode uses a helper program to send packets to the target of the port forward. It is useful in setups where the interface and/or gateway IP used for communication with the target cannot be accurately determined at the time the rules are loaded. 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 and UDP protocols are supported.

The pure NAT 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 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.

Individual NAT rules also contain an option to override the behavior determined by this selection, so they may have NAT reflection forced on or off on a case-by-case basis.

### Reflection Timeout

The Reflection Timeout setting allows you to force a timeout for connections made when performing NAT reflection for port forwards in NAT + Proxy mode. If some connections are staying open longer than desired and causing issues, this option would help mitigate that issue.

### NAT Reflection for 1:1 NAT

When checked, this option adds additional NAT 1:1 mappings for access to 1:1 mappings of your external IP addresses from within your internal networks. This gives the same functionality that already exists for port forwards, but for 1:1 NAT. There are some complex routing scenarios that may render this option ineffective. Additionally, this 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.

### Outbound NAT for 1:1 NAT Reflection

When the box is checked for Automatically create outbound NAT rules which assist inbound NAT rules that direct traffic back out to the same subnet it originated from, it does what the option says. It adds some additional rules which allow 1:1 NAT Reflection to function fully. In most cases, you will want to check this box if you want 1:1 NAT Reflection to work. This only works for assigned interfaces. Other interfaces require manually creating the outbound NAT rules that direct the reply packets back through the router.

### TFTP Proxy

The built-in TFTP proxy will proxy connections to TFTP servers that are outside the firewall, so that client connections may be made to remote TFTP servers. You may ctrl-click or shift-click to select multiple entries from the list. If no interfaces are chosen, the TFTP proxy service is deactivated.

# Networking Tab

## IPv6 Options

### Allow IPv6

Starting with pfSense 2.1, IPv6 filtering is supported so the Allow IPv6 Traffic option is enabled by default for configurations that started on version 2.1. If you do not want to allow IPv6 traffic through your firewall, you may uncheck this box and all IPv6 traffic will then be blocked and logged if you are logging packets dropped by the default deny rule. Systems upgraded from earlier versions retain the value of this setting from prior to the upgrade, which defaulted to not allowing IPv6.

### IPv6 over IPv4 Tunneling

You may also check Enable IPv4 NAT encapsulation of IPv6 packets by checking that box, which enables IP protocol 41/RFC 2893 forwarding to the IPv4 address specified in the IP address field. That would allow you to forward all IPv6 traffic to a host behind this firewall instead of handling it locally.

## Network Interfaces

### Device Polling

Device polling is a technique that lets the system periodically poll network devices for new data instead of relying on interrupts. This prevents your WebGUI, SSH, etc. from being inaccessible due to interrupt floods when under extreme load, at the cost of slightly higher latency (up to 1 ms). This is usually unnecessary, unless your hardware is undersized.

Polling also requires hardware support in your system's network cards. According to the polling(4) man page for FreeBSD 8.3 (upon which pfSense 2.1 is based), the `bge(4)`, `dc(4)`, `em(4)`, `fwe(4)`, `fwip(4)`, `fxp(4)`, `ixgb(4)`, `nfe(4)`, `nge(4)`, `re(4)`, `rl(4)`, `sf(4)`, `sis(4)`, `ste(4)`, `stge(4)`, `vge(4)`, `vr(4)`, and `xl(4)` devices are supported, with support for others pending in future FreeBSD releases.

**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

Checking this option will disable hardware checksum offloading. Checksum offloading is broken in some hardware, particularly some Realtek cards. Rarely, drivers may have problems with checksum offloading and some specific NICs. Typical symptoms of broken checksum offloading include corrupted packets and poor throughput performance.

### Hardware TCP Segmentation Offloading

Checking this option will disable hardware TCP segmentation offloading (TSO, TSO4, TSO6). This offloading is broken in some hardware drivers, and may impact performance with some specific NICs. This option is more desirable for workstations/appliances than for routers and firewalls, so it is left as an option in case it does increase performance for certain deployments.

### Hardware Large Receive Offloading

Checking this option will disable hardware large receive offloading (LRO). This offloading is broken in some hardware drivers, and may impact performance with some specific NICs. As with the TSO option above, this option is more desirable for workstations/appliances than for routers and firewalls, so it is left as an option in case it does increase performance for certain deployments.

### Suppress ARP messages

If you have two or more interfaces which share the same physical network, such as in a scenario where multiple interfaces are plugged into the same broadcast domain, this option will hide the spurious ARP messages that would otherwise overload the logs with useless entries.

# Miscellaneous Tab

# Proxy Support

If this firewall resides in a network behind a proxy, you may enter the proxy options below so that requests from the firewall for things 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.

## 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.

## Proxy Username

If required, this is the username that should be sent when the firewall must use the proxy.

## Proxy Password

If required, this is the password associated with the username set in the previous option.

# Load Balancing

## Sticky Connections

The text in the WebGUI best explains Sticky Connections option in this section: Successive connections will be redirected to the servers in a round-robin manner with connections from the same source being sent to the same web server. This "sticky connection" will exist as long as there are states that refer to this connection. Once the states expire, so will the sticky connection. Further connections from that host will be redirected to the next web server in the round robin.

"Sticky" connections are desirable for some applications that rely on the same IPs being maintained throughout a given session. This is used in combination with the server load balancing functionality, described further in Chapter 22, *Server Load Balancing*.

This option applies to outbound load balancing (Multi-WAN) as well as server load balancing.

You may also enter a value for the source tracking timeout of sticky connections. This will retain the sticky association from a host after the all of the states from that host expire for however many seconds are entered in the box. By default, this value is not set, which assumes 0 for the value, deleting the association as soon as the states expire. If you find that sticky is working for you, but seems to stop working partway through sessions, you can increase this value to hold an association longer.

## Gateway Switching

The Allow default gateway switching option, disabled by default, will allow other gateways to take over should the default gateway become unreachable. With multiple WANs, this would ensure that the firewall always has a default gateway so that traffic from the firewall itself can always get out to the Internet. There are many cases where this is not desirable, however, such as when you have an additional "WAN" that is not connected to the Internet. In the future this option will be expanded so it can be controlled on a per-gateway basis.

# Power Savings

If the Use PowerD option is checked, then the **powerd** daemon is enabled. This daemon monitors the system and can lower the CPU frequency during periods of low 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. 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.

The mode for **powerd** may also be selected. Four modes exist, maximum, minimum, adaptive, and hiadaptive. Maximum keeps the performance as high as possible. 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 is 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.

You may select one mode for normal operation and another for when the system is on battery power. Support for battery power detection varies by hardware.

## Cryptographic Hardware Acceleration

There are a few options available for accelerating cryptographic operations via hardware. Some are built into the kernel, and others are loadable modules. Two of the optional modules are selectable here: The AMD Geode LX Security Block (`glxsb`) and AES-NI (Advanced Encryption Standard, New Instructions).

The `glxsb` driver is used mainly in ALIX and Soekris embedded systems. It is a cryptographic accelerator which can improve performance for certain ciphers, such as AES-128. This can improve VPN performance and other subsystems which may use AES-128, such as SSH. If you select AMD Geode LX Security Block (glxsb), the driver will be loaded at startup so that the accelerator chip may be used. This driver can conflict with other cryptographic accelerator cards, such as those from Hifn, and take precedence over them when both are found. If you have a Hifn card, you should uncheck this option so that the `glxsb` device is never loaded. If the driver is already in use, you must reboot after setting this option so it can be unloaded.

If AES-NI CPU-based Acceleration (aesni) is chosen, then its kernel module will be loaded when saved, and at bootup. As with `glxsb`, `aesni` will accelerate certain AES-based ciphers. Support for AES-NI is found in many recent Intel CPUs and some AMD CPUs. Intel CPU support can be found in Westmere (except i3), Sandy Bridge (except i3, Celeron, Pentium), and Ivy Bridge (all variants have it) CPUs. AMD support is limited to their Bulldozer line as of this writing. Speeds with AES-NI vary by support of the underlying software as well. Some OpenSSL-based software like OpenVPN can perform differently with AES-NI unloaded since OpenSSL has built-in support for AES-NI.

These drivers hook into FreeBSD's `crypto(9)` framework, so many aspects of the system will automatically use the feature for supported ciphers. For OpenVPN to use these accelerators, on the VPN settings, find the Hardware Crypto field and set it to BSD cryptodev engine.

There are other supported cryptographic devices, such as `hifn(4)` (see Table 3.4, “IPsec Throughput by Cipher — ALIX”), `ubsec(4)`, and VIA `padlock(4)`. In most cases, if a supported accelerator chip is detected, it will be shown in the System Information widget.

## Thermal Sensors

pfSense 2.1 can read temperature data from a few sources to display on the dashboard. If you have a supported CPU, selecting a thermal sensor will load the appropriate driver to read its temperature. Setting this to **None** will attempt to read the temperature from an ACPI-compliant motherboard sensor instead, if one is present.

The **coretemp** module supports reading thermal data from Intel core-series CPUs and other modern Intel CPUs using their on-die sensors.

The **amdtemp** module supports reading thermal data from AMD K8, K10, K11 and other modern AMD CPUs using their on-die sensors.

If you do not have a supported thermal sensor chip in your system, this option will have no effect. To unload the selected module, set this option to 'none' and then reboot.

### Note

The **coretemp** and **amdtemp** modules report the thermal data from the CPU core itself. This may or may not be indicative of the temperature elsewhere in the system. Case temperatures can vary from place to place. Some motherboards may have multiple sensors that can be read by other programs such as **mbmon**, though support for those is not in the GUI currently.

## IP Security

### Security Associations

By default, if several IPsec Security Associations (SA) match, the newest one is preferred if it's at least 30 seconds old. Select this option to always prefer old SAs over new ones. This is rarely desirable, except when dealing with some specific third-party devices. For more on Security Associations, refer to the section called “Security Association”

### IPsec Debug

If Start racoon in debug mode is checked, the IPsec daemon (**racoon**) will be launched in debug mode. This mode will produce extremely verbose logs, which can be useful for tracking down issues with IPsec tunnel negotiation. If you change this option, when you save the settings on this page, the racoon daemon will be restarted, which will drop all active IPsec tunnels.

### IPsec Reload on Failover

In some circumstances using a gateway group as the interface for an IPsec tunnel does not function properly, and IPsec must be forcefully reloaded when a failover occurs. Because this will disrupt all IPsec tunnels, this behavior is disabled by default. Check this box to force IPsec to fully reload on failover. This also will work for a WAN failure in a single WAN scenario also if you have issues with a tunnel re-establishing after your WAN disconnects and reconnects.

### Maximum MSS

The Enable MSS clamping on VPN traffic option will enable MSS clamping on TCP flows over VPN. This helps overcome problems with PMTUD on IPsec VPN links. If left blank, the default value is 1400 bytes. Without this option, TCP packets going over IPsec would be fragmented and prone to loss if PMTUD does not result in their size being reduced automatically.

## Schedules

This option controls whether or not the states are cleared when a scheduled rule transitions into a state that would block traffic. If checked, connections are terminated when the schedule time has expired. If unchecked, 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, the monitoring process will flush states for a gateway that goes into a down state. This option overrides that behavior by not clearing states for existing connections, leaving them to timeout on their own or in some cases, continue even if a WAN's quality is degraded but still usable.

More information on how this impacts Multi-WAN can be found in the section called “State Killing/ Forced Switch”.

### Skip Rules When Gateway is Down

By default, when a rule has a specific gateway set and this gateway is down, a rule is created and traffic is sent to the default gateway. This option overrides that behavior and the rule is not created when gateway is down, so instead of flowing via the default gateway, the traffic will continue to attempt to use the gateway that is in a down state, and it will most likely not proceed. This is useful if you have traffic that should only ever use one specific WAN and never flow over any other WAN, regardless of how the firewall's routing table has for the default route.

## RAM Disk Settings

The `/tmp` and `/var` directories are very frequently used for writing files and keeping data that is temporary and/or volatile. On NanoBSD, and formerly embedded, these have always been kept in a RAM disk to reduce the amount of writing that needed to happen on the CF. Today, with the advent of SSDs, users frequently want the best of both worlds: They want to use their larger SSD for more tasks as a full installation, but they are also concerned with frequent disk writes.

pfSense 2.1 introduces a new feature that lets a full install of pfSense also use a RAM disk for `/tmp` and `/var`, to reduce the amount of writing done to the install disk. This 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.

To start using a RAM disk, check the option Use memory file system for /tmp and /var, and then enter disk sizes for `/tmp` and `/var`.

### Caveats/Precautions

There are several things one needs to be cautious about when choosing whether or not to use the RAM disk option on a full install. It is possible to have data loss or other unexpected failures due to the use of this option.

Because the RRD and DHCP lease databases are kept in `/var` this also means that a full install using this option can lose that data if there is a sudden power loss. Later in this section we discuss how to setup periodic backups of that data.

The system logs are also held in `/var` but they are not backed up like the RRD and DHCP databases. The logs will reset fresh on each reboot.

Since these are RAM disks, the amount of RAM available to other programs will be reduced by the amount of space used by the RAM disks. For example if you have 1GB of RAM, and set 256MB for `/var` and 256MB for `/tmp`, then only 512MB of RAM will be available to the OS for general use.

Special care must be taken when choosing a RAM disk size, more discussion on that will be done in the next 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 you will want much larger sizes if you have enough RAM. The most common failure is that when you install a package, 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.

On ALIX hardware, there is really not enough RAM to tweak these sizes in any meaningful increment.

For `/tmp`, a minimum of 40MB is required. For `/var` a minimum of 60MB is required. If you need some help determining the size, then you can check the current usage of your `/tmp` and `/var`

directories before making a switch. Check the usage several times over the course of a couple days so you don't accidentally catch it at a low point. Watching the usage during a package installation would add another useful data point.

## Periodic Data Backups

When using RAM disks, either on NanoBSD or on a full install with the RAM disk option enabled, temporary data that needs written often is kept on a RAM disk and then saved during a normal shutdown or reboot. This data can be lost should you suffer a sudden power loss. The two options in this section, RRD Backup and DHCP Leases backup, let you configure a periodic backup of the RRD graph databases and the DHCP lease database, respectively. You can choose any hourly option between one hour and 24 hours.

### Note

The more frequent the backup schedule, the more writes happen to the media, so if you are worried about wearing out the media use a longer term value. However, a longer term value is more likely to lose data in the event of a power loss. Consider how important the data is with respect to your media's lifetime before enabling these backups.

# System Tunables Tab

The System Tunables tab provides a means to set run-time FreeBSD system tunables, also known as **sysctl** OIDs. In almost all cases, these should be left at their defaults. Those familiar with FreeBSD, or 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 values here are distinct from values that are considered Loader Tunables. Loader Tunables are read-only values once the system has been booted, and to change those values they must be set in `/boot/loader.conf.local` or `/boot/loader.conf`.

# Notifications

Starting with pfSense 2.0, the firewall is now capable of sending remote notifications by using Growl or E-mail via SMTP. These notifications are the same notifications that will pop up in the GUI as a scrolling notice in the top area of the page. The exact location and color of the display varies based on the theme.

## Growl

Growl provides a fairly unobtrusive method of delivering desktop notifications. These notifications pop up on the desktop and then hide or fade away. Growl is built-in on Mac OSX, and it's available with additional software on Windows [http://www.growlforwindows.com/gfw/default.aspx] and FreeBSD/Linux (Gnome [http://the.taoofmac.com/space/projects/DBUSGrowl] or KDE [http://www.pingle.org/2011/04/15/growl-server-kde]).

### Disable Growl Notifications

When checked, Growl notifications will not be sent, but the settings will be preserved.

### Registration Name

This is the service name used to register with the Growl server. By default this is *`PHP-Growl`*. Consider this as the type of notification, as seen by the Growl server.

### Notification Name

The name of the system producing the notification. The default value of *pfSense growl alert* may be sufficient, but you may customize it with the firewall's hostname or any other value to make it distinct.

### IP Address

The IP address to which the Growl notifications will be sent.

### Password

The password required by the Growl server to deliver notifications.

### Test Growl

The Test Growl button is used to send a test notification via Growl using the current settings.

## SMTP E-mail

E-mail notifications are delivered by a direct SMTP connection to a server of your choice. The server should either be configured to allow relaying from the firewall, or you may use the configuration options below to provide credentials for SMTP authentication.

### Disable SMTP Notifications

When checked, SMTP notifications will not be sent, but the settings will be saved. This is useful if you do not want notifications, but you do need to have the settings in place for a package such as mailreports which can generate periodic reports and send them via e-mail.

### E-mail server

Place 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 for communicating with the SMTP server. The most common ports are 25, 587, and 465. In most cases, 25 may work. Some providers block outbound connections to port 25, so using 587 (the Submission port) is preferred if your server supports it. Port 465 is for Secure SMTP (smtps), so using that port will likely also require checking Enable SSL/TLS Authentication. SSL/TLS Authentication may be used on any port if the server supports SSL/TLS, but it is required on port 465.

### From e-mail address

This is the e-mail address that will be used as the source of the e-mail in the From header. You may set this to a nonexistent e-mail address, or the same as the destination e-mail address, or some other value. Some SMTP servers attempt to validate this address so you may need to use a valid address here if the e-mail will be routed through a strict server.

### Notification E-mail address

This is the destination address of the notification e-mail. It can be your own personal address, a general support account, or any other address you want to receive the notifications.

### Notification E-Mail auth username

If your server requires a username and password for SSL/TLS Authentication, you may enter the username here.

### Notification E-Mail auth password

If your server requires a username and password for SSL/TLS Authentication, you may enter the password here.

### Test SMTP

The Test SMTP button will generate a test notification and send it via SMTP using the current settings.

## Startup/Shutdown Sound

If the firewall hardware has a PC speaker, normally pfSense will play a sound when startup finishes, and again when a shutdown is being initiated. By checking Disable the startup/shutdown beep, these sounds will not be played.

# Console Menu Basics

Some configuration and maintenance tasks may also be performed from the system console. The console may be reached by using the keyboard and mouse, serial console if enabled or using embedded, or by using SSH. 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 2.1-RELEASE-pfSense (amd64) on pfsense ***

 WAN (wan)       -> em0        -> v4: 1.2.3.4/24
                                  v6: fd01::6/64
 LAN (lan)       -> em1        -> v4: 192.168.28.1/24
                                  v6: fd05::1/64

 0) Logout (SSH only)                  8) Shell
 1) Assign Interfaces                  9) pfTop
 2) Set interface(s) IP address       10) Filter Logs
 3) Reset webConfigurator password    11) Restart webConfigurator
 4) Reset to factory defaults         12) pfSense Developer Shell
 5) Reboot system                     13) Upgrade from console
 6) Halt system                       14) Disable Secure Shell (sshd)
 7) Ping host                         15) Restore recent configuration
```

What follows is a general description of what is possible by using most of these options. As with other advanced options, some of these may be covered with more detail in other sections of the book where their discussion would be more topical or relevant.

# Assign Interfaces

This will restart the Interface Assignment task, which was covered in detail in the section called “Assigning Interfaces” and the section called “Manually Assigning Interfaces”. You can create VLAN interfaces, reassign existing interfaces, or assign new ones.

# Set interface(s) IP address

This option can be used in the obvious manner, to set the WAN, LAN, or OPT interface IP address, but there are also some other useful tasks that happen when resetting the LAN IP. For starters, when this is set, you also get the option of turning DHCP on or off, and setting the DHCP IP range.

If you have disabled the WebGUI anti-lockout rule, you will be prompted to re-enable it. It will also prompt to revert to HTTP on the default port if using a non-standard port. This is done to help those who may find themselves locked out from using the WebGUI regain access.

# Reset webConfigurator password

This option will reset the WebGUI username and password back to **admin** and **pfsense**, respectively.

# Reset to factory defaults

This will restore the system configuration back to factory defaults. Be aware that this will not, however, make any changes to the filesystem or the packages installed on the OS. If you suspect that system files have been corrupted or altered in some undesirable way, the best practice is to make a backup, and reinstall from CD or other installation media. (Also possible in the WebGUI at Diagnostics → Factory defaults)

# Reboot system

This will cleanly shutdown the pfSense system and restart the OS (Diagnostics → Reboot in the WebGUI).

# Halt system

This will cleanly shutdown the system and either halt or power off, depending on hardware support. It is not recommended to ever pull the plug out of a running system, even embedded systems. Halting before removing power is always the safest choice should you ever need to turn off the system. On embedded systems, pulling the plug is less dangerous, but if the timing is bad it could also be harmful (Diagnostics → Halt System in the WebGUI).

# Ping host

Prompts for an IP address, which will be sent three ICMP echo requests. The output from the **ping** will be shown, including the number of packets received, sequence numbers, response times, and packet loss percentage. If you enter an IPv4 address or a hostname, **ping** will be run. If you enter an IPv6 address, **ping6** will be run.

# Shell

Starts a command line shell. Very useful, and very powerful, but also has the potential to be very dangerous. Some complex configuration tasks may require working in the shell, and some troubleshooting tasks are easier to accomplish from here, but there is always a chance of causing irreparable harm to the system if not handled with care. The majority of pfSense users may never touch the shell, or even know it exists.

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 reasons of security and size constraints.

The shell started in this manner will be **tcsh**, and the only other shell available is **sh**. While it may be possible to install other shells for the convenience of those who are very familiar with the OS (see the section called “Using Software from FreeBSD's Ports System (Packages)”), this is not recommended or supported.

# PFtop

PFtop gives you a real-time view of the firewall states, and the amount of data they have sent and received. It can help pinpoint what IP addresses and sessions are currently using bandwidth, and may

also help diagnose other network connection issues. See the section called “Viewing with pftop” for more details.

# Filter Logs

Using the Filter Logs option, you will see any filter log entries appear in real-time, in their raw form. There is quite a bit more information shown per line than you will typically see in the firewall log view in the WebGUI (Status → System Logs, Firewall tab), but not all of this information is easy to read.

# Restart webConfigurator

Restarting the webConfigurator will restart the system process that runs the WebGUI. On rare occasions there may be a change that might need this before it will take effect, or in extremely rare cases the process may have stopped for some reason, and restarting it will restore access.

If for some reason the GUI is not responding and this option does not restore access, you can attempt to run the same sequence of commands manually from the shell:

```
# killall -9 php; killall -9 lighttpd; /etc/rc.restart_webgui
```

# pfSense Developer Shell (Formerly PHP shell)

The Developer shell, which used to be known as the pfSense PHP shell, is a very powerful utility that lets you execute PHP code in the context of the running system. As with the normal shell, is can also be very dangerous to use, and easy for things to go wrong. This is mainly 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 pfSense Developer Shell that can automate some simple tasks that might be necessary to perform from the console if the GUI is unreachable.

These scripts are run from within the shell like so:

```
pfSense shell: playback scriptname
```

They may also be run from the command line:

```
# pfSsh.php playback scriptname
```

### disabledhcpd

This script removes all DHCP configuration from the firewall's configuration, effectively disabling the DHCP service and completely removing all traces of its settings.

### disablereferercheck

This script disables the HTTP_REFERER check mentioned in the section called “Browser HTTP_REFERER enforcement”. This can help gain access to the GUI if your browser session is triggering this protection.

### enableallowallwan

This script adds an allow all rule for IPv4 and IPv6 to the WAN interface. Be extremely careful with this option, it is meant to only be a temporary measure to gain access to services on the WAN interface of the firewall, such as the GUI, were more precise rules can be added. Once you have a rule set to allow access to the GUI as needed, remove the allow all rules added by this script.

### enablesshd

This script enables SSHD, 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.

### gitsync

This is a complex script that will synchronize the PHP and other script sources with the files from the pfSense git repository. It is most useful on development snapshots to pick up changes from more recent commits. It can be dangerous to use in other circumstances, so only use this under the direction of a knowledgeable developer or support representative. If you run the command without any parameters, it will print a help message outline its use. More information can be found on the pfSense Doc Wiki [http://doc.pfsense.org/index.php/Updating_pfSense_code_between_snapshots].

### removepkgconfig

This script will remove all traces of package configuration 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.

### restartdhcpd

This script will stop and restart the DHCP daemon.

### restartipsec

This script will stop and restart racoon, the IPsec daemon.

## Upgrade from console

Using this option, it is possible to upgrade by entering a full URL to a pfSense firmware image, or a full local path to an image uploaded in some other manner. This method of upgrading is covered in more detail in the section called “Upgrading using the Console”.

## Enable/Disable Secure Shell (sshd)

This option will allow you to toggle the status of the Secure Shell daemon, sshd. It works similarly to the same option in the WebGUI covered earlier in this chapter, but is accessible from the console.

## Restore recent configuration

This menu option will start a script that can list and restore backups from the configuration history. This is similar to accessing the configuration history from the GUI at Diagnostics → Backup/Restore. You can view the last few configuration files, along with a timestamp and description of the change made in the config, the user and IP that made the change, and the config revision. This is especially useful if a recent configuration error accidentally removed your access to the GUI.

## Move configuration file to removable device

If you wish to keep your system configuration on removable storage, such as a USB thumb drive, this option can be used to relocate the configuration file. Once used, be sure to make sure the media is accessible at boot time so that it may be reloaded. This is not a normal method of backing up the configuration. For information on making backups, see Chapter 9, *Backup and Recovery*.

# Time Synchronization

Time and clock issues are not that uncommon when configuring any system, but they can be important to get right on firewalls, especially if they are performing any kind of tasks involving validating certificates as part of a PKI infrastructure. Getting time synchronization to work properly is also 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. There can be some quirks to getting not only a proper date and time into the system, and keeping it that way, but also in making sure that the time zone is properly reflected.

Not only will getting this all in line help with critical system tasks, but it also ensures that your log files are properly timestamped, which can greatly aid in troubleshooting, record keeping, and general system management.

# Time Keeping Problems

You may run into hardware that has significant problems keeping time. All PC clocks will drift to some extent, but you may find some hardware that will drift as much as one minute for every couple minutes that pass and get wildly out of sync quickly. 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 should you encounter it, the following will outline the things that usually fix this.

There are four things to check if you encounter hardware with significant time keeping problems.

## Network Time Protocol

By default, pfSense is configured to synchronize its time using the ntp.org Network Time Protocol (NTP) server pool. This ensures an accurate date and time on your system, and will accommodate normal clock drift. If your system's 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, the 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.

## 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 for some reason. Applying the BIOS update fixed the problem. The first thing you should check is to make sure you have the latest BIOS on your system.

## PNP OS settings in BIOS

I have encountered other hardware that had time keeping difficulties in FreeBSD and pfSense unless PNP OS in the BIOS was set to "No". If your BIOS does not have a PNP OS configuration option, look for an "OS" setting and set it to "Other".

## Disable ACPI

Some 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 systems that would not boot or run properly unless ACPI support was disabled in the BIOS and/or in the OS.

The best way to disable ACPI is in the BIOS. If there is no BIOS option to disable ACPI, then you can try to run without it in two different ways. The first, temporary method is to disable ACPI at the

boot prompt. Early in the boot process, a menu appears with several choices, one of which is Boot pfSense with ACPI disabled. By choosing this, ACPI will be disabled for this single boot. If behavior improves, then you should disable ACPI permanently.

To permanently disable ACPI, you must add a setting to the `/boot/device.hints` file. You can do this by browsing to Diagnostics → Edit File, enter `/boot/device.hints` and then click Load. Add a new line at the end and then enter:

```
hint.acpi.0.disabled="1"
```

Then click Save.

For an alternate way to do this, from Diagnostics → Command or from a shell, type this:

```
# echo "hint.acpi.0.disabled=1" >> /boot/device.hints
```

### Note

The `/boot/device.hints` file will be overwritten during an upgrade. Be aware that you will need to repeat this change after performing a firmware update.

## Adjust Timecounter Hardware Setting

On very few 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 VMware ESX 5.0 in combination with an amd64-based pfSense image (or any FreeBSD amd64 image.). Its cause is a bug in the hypervisor that is fixed in 5.1. There is a work around if you cannot upgrade ESX. On these systems, using the default timecounter, the clock will stop 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 timecouter temporarily, browse to Diagnostics → Command and execute the following:

```
# sysctl -w kern.timecounter.hardware=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, you need to make this change permanent. The previously made change will not survive a reboot. Browse to System → Advanced, go to the System Tunables tab, and add a new entry and set the Tunable to **`kern.timecounter.hardware`** and the Value to **`i8254`**.

Click Save, and then that setting should be read back in on the next boot.

Depending on your platform and hardware, there may also be other timecounters to try. For a list of available timecounters found on your system, execute the following command:

```
# sysctl kern.timecounter.choice
```

You should then see a list of available timecounters and their "quality" as reported by FreeBSD:

```
kern.timecounter.choice: TSC(-100) ACPI-safe(850) i8254(0) dummy(-1000000)
```

You could then attempt to 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. The TSC is a counter on the CPU, but is tied to the clock rate and is not readable by other CPUs. This makes its use in SMP systems impossible, and in those with variable-speed CPUs. The i8254 is a clock chip found in most hardware, which tends to be safe but can have some performance drawbacks. The ACPI-safe counter, if properly supported in the available hardware, 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. This and more information on

FreeBSD Timecounters can be found in the paper *Timecounters: Efficient and precise timekeeping in SMP kernels [http://phk.freebsd.dk/pubs/timecounter.pdf]* by Poul-Henning Kamp of the FreeBSD Project.

### Adjust the Kernel Timer Frequency

In some cases it may also be necessary to adjust the kernel timer frequency, or kern.hz kernel tunable. 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 to 100, which should work fine in nearly all cases with VMware products. As with the timecounter setting above, to adjust this setting you add a line to `/boot/loader.conf` with the new value:

```
kern.hz=100
```

## GPS Time Synchronization

To aid in maintaining an accurate clock, GPS time synchronization was added in pfSense 2.1. 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 the section called “NTPD”.

# Troubleshooting

The Setup Wizard and related configuration tasks will work for most, but there may be some issues getting packets to flow normally in their intended directions. Some of these issues may be unique to your particular setup, but can be worked through with some basic troubleshooting.

## Cannot access WebGUI from LAN

The first thing to check if you cannot access the WebGUI from the LAN is the cabling. If you are directly connecting a client PC to a network interface on a pfSense system, you may need a crossover cable unless one or both network cards support Auto-MDIX.

Once you are sure there is a link light on both the client's network card and the pfSense LAN interface, the next thing to check is the TCP/IP configuration on the PC from which you are trying to connect. If the DHCP server is enabled on the pfSense system, as it will be by default, ensure that the client is also set for DHCP. If DHCP is disabled on the pfSense system, you will need to hard code an IP address on the client residing in the same subnet as the pfSense system's LAN IP address, with the same subnet mask, and use the pfSense LAN IP address as its gateway and DNS server.

If the cabling and network settings are correct, you should be able to ping the LAN IP of the pfSense system from the client PC. If you can ping, but you are still unable to access the WebGUI, there are still a few more things to try. First, if the error you receive on the client PC is a connection *reset* or *failure*, then either the server daemon that runs the WebGUI is not running, or you are trying to access it from the wrong port. If the error you receive is instead a connection *timeout*, that points more toward a firewall rule.

If you receive a connection reset, you may first try to restart the WebGUI server process from the system console, typically option 11. Should that not help, start a shell from the console (option 8), and type:

```
# sockstat | grep lighttpd
```

That should return a list of all running lighttpd processes, and the port upon which they are listening, like so:

```
root     lighttpd   33098 11 tcp4   *:443                 *:*
root     lighttpd   33098 12 tcp6   *:443                 *:*
```

```
root     lighttpd   33098 13 tcp4   *: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 on IPv4 for the redirect, but that may vary based on your configuration. Try connecting to the pfSense LAN IP by using that port directly, and with both HTTP and HTTPS. For example, if your LAN IP 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`**.

If you receive a connection timeout, refer to the section called “What to do if you get 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.

It is also a good idea to 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 router, and try again.

# No Internet from LAN

If you are 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 that pfSense sees it as operational. Browse to Status → Interfaces, and look at the WAN interface status there. The status should show as "up". If it shows down, double check your cabling and WAN settings under Interfaces → WAN. If you are 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 your settings on Interfaces → WAN, check or reboot your ISP equipment (cable/DSL modem, etc.), and perhaps consult with your ISP for help regarding the settings you should use there.

## DNS Resolution Issues

Inside the WebGUI, go to Diagnostics → Ping, and enter in your ISP's gateway address if you know it. It will be listed on Status → Interfaces for the WAN interface. If you do not know the gateway, you may try some other known-valid address such as **`4.2.2.2`**. If you are able to ping that address, then repeat that same ping test from your client PC. Open a command prompt or terminal window, and ping that same IP address. If you can ping that IP address, then try to ping a site by name such as **`www.google.com`**. Try it from the pfSense WebGUI and from the client PC. If the IP ping test works, but you cannot ping by name, then there is a problem with DNS resolution. (See Figure 10.23, “Testing connectivity for bogon updates” for an example.)

If DNS resolution does not work on the pfSense system, check your DNS server settings under System → General Setup, and under Status → Interfaces. Check with ping to be sure they are reachable. If you can reach the gateway address at your ISP, but not their DNS servers, it may be advisable to call your ISP and double check those values. If your DNS servers are obtained via DHCP or PPPoE and you cannot contact them, you may also need to contact your ISP regarding that issue. If all else fails, you may want to consider using OpenDNS [http://www.opendns.com/] (see the section called “Free Content Filtering with OpenDNS”) or Google's public DNS (8.8.8.8, 8.8.4.4) name servers on your pfSense firewall instead of those provided by your ISP.

If DNS works from the pfSense router, but not from a client PC, it could be the DNS Forwarder configuration on the pfSense system, the client configuration, or firewall rules. Out of the box, pfSense has a DNS forwarder which will handle DNS queries for clients behind the router. If your client PCs are configured with DHCP, they will be getting the IP address of the pfSense router interface to which

they are connected as a DNS server, unless you specify an override. For example, if a PC is on the LAN side, and the pfSense system's LAN IP address is 192.168.1.1, then the client's DNS server should also be 192.168.1.1. If you have disabled the DNS Forwarder, you may also need to adjust the DNS servers which get assigned to DHCP clients under Services → DHCP Server. Normally when the DNS Forwarder is disabled, the system's DNS servers are assigned directly to the clients, but if that is not the case in practice for your setup, define them here. 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 system or whatever internal or external DNS servers you would like for it to use.

Another possibility for DNS working from pfSense itself but not a local client is an overly strict firewall rule. Check Status → System Logs, on the Firewall tab. If you see blocked connections from your local client trying to reach a DNS server, then you should add a firewall rule at the top of the ruleset for that interface which will allow connections to the DNS servers on TCP and UDP port 53.

## Client Gateway Issue

In order for the pfSense system to properly route Internet traffic for your client PCs, it must be their gateway. If the client PCs are configured using pfSense's DHCP server, 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 system. For example, if the clients are on the pfSense LAN side, and the IP address for pfSense's LAN interface is 192.168.1.1, then a client's gateway address 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 router may be blocked. This should be 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 your LAN ruleset at Firewall → Rules, then the LAN tab and make the necessary adjustments to allow that traffic. Consult Chapter 10, *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 you have rules in place to allow the traffic to pass. No rule is created by default on OPT interfaces.

## NAT Rule Issues

If the outbound NAT rules have been changed from their defaults, it may also be possible that traffic attempting to reach the Internet does not have NAT properly applied. Navigate to Firewall → NAT, and go to the Outbound tab. Unless you are sure that you need it set to manual, change the setting to Automatic outbound NAT rule generation (IPsec passthrough) 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 you have this set to Manual Outbound NAT rule generation (Advanced Outbound NAT (AON)), and it works from LAN but not from an OPT interface, you will need to manually set a rule that matches traffic coming from there. Look at the existing rule for LAN and adjust it accordingly, or refer to the NAT chapter for more information on creating outbound NAT rules. The same applies for traffic coming from VPN users: PPTP, OpenVPN, IPsec, etc. If these users need to reach the Internet via this pfSense router, they will need outbound NAT rules for their subnets. See the section called “Outbound NAT” for more information.

# pfSense's XML Configuration File

pfSense stores all of its 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.

Some people who are familiar with FreeBSD and related operating systems have found this out the hard way, when their changes to some system configuration files were repeatedly overwritten by the system 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 your configuration

A few configuration options are only available by manually editing your 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.

The safest and easiest method of editing the configuration file is to make a backup from Diagnostics → Backup/Restore, save the file to your PC, edit the file and make any needed changes, then restore the altered configuration file to the system.

If you are familiar with the **vi** editor, and you are extremely careful, the **viconfig** command will edit the running configuration live, and when you save and quit, it will remove the cached configuration from `/tmp/config.cache` and then the changes should be visible in the GUI, and will be active next time the service relevant to the edited portion of the config is restarted/reloaded.

# What to do if you get locked out of the WebGUI

Under certain circumstances you may find yourself locked out of the WebGUI, mostly due to pilot error. Don't be afraid if this happens to you; there are a number of ways to get back into the firewall GUI. Some methods are a little tricky, but it should always be possible to regain access. The worst-case scenarios require physical access. As you'll remember from earlier this chapter we mentioned that anyone with physical access can bypass security measures and now you will see just how easy it is.

## Forgotten Password

If you forgot the password for the system it can be reset easily with console access. Get to the physical console (Keyboard/Monitor, or Serial) and use option **3** to reset the WebGUI password.

## Forgotten Password with a Locked Console

If the console is password protected and you do not know the password, all is not lost. It will take a couple reboots to accomplish, but it can be fixed with physical access to the console:

- Reboot the pfSense box
- Choose option 5 (Single User Mode) from the loader menu (the one with the ASCII pfSense logo)
- Press enter when prompted to start /bin/sh
- Remount all of the 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

You should now be able to access the system with the default username and password of **admin** and **pfsense**, respectively.

# HTTP vs HTTPS Confusion

Ensure you are connecting with the proper protocol, either HTTP or HTTPS. If one doesn't work, try the other. You may find that you need to try the opposite protocol on the others port, like so:

- **http://pfsensebox:443**
- **https://pfsensebox:80**

If you need to reset this from the console, reset the LAN IP, enter the same IP, and it will prompt to reset the WebGUI back to HTTP.

# Blocked Access with Firewall Rules

If you blocked yourself out of the WebGUI remotely with a firewall rule, there may still be hope. This can't happen from the LAN unless you disable the anti-lockout rule that maintains access to the WebGUI from that interface.

Having to walk someone on-site through fixing the rule is better than losing everything!

# Remotely Circumvent Firewall Lockout with Rules

There are a few ways you can manipulate the firewall behavior at the shell that can get you back into 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 you know the IP address you're attempting to access the firewall from, is to use the **easyrule** shell command to add a new firewall rule to get you in to the GUI. In the following example, it will allow access on the *WAN* interface, from *x.x.x.x* (the client IP) to *y.y.y.y* (presumably the WAN IP) on TCP port *443*.

```
# easyrule pass wan tcp x.x.x.x y.y.y.y 443
```

Once the rule has been added, you should then be able to access the GUI.

## Add an allow all WAN rule from the shell

Another tactic is to temporarily activate an "allow all" rule on the WAN to let you in by running this command at the shell:

```
# pfSsh.php playback enableallowallwan
```

Once you have fixed the rules and regained access, remove the "allow all rule" on WAN.

## Disable the Firewall

You could (very temporarily) disable firewall rules by using the console. You can use the physical console, or if you are still able to get in via SSH, that will also work. From the console, use option 8 to start a shell, and then type:

```
# pfctl -d
```

That will disable the firewall, including all NAT functions. You should then be able to get into the WebGUI from anywhere, at least for a few minutes or until you save something in the WebGUI that causes the ruleset to be reloaded (which is almost every page). Once you have adjusted the rules and regained the necessary access, turn the firewall back on by typing:

```
# pfctl -e
```

## Manual Ruleset Editing

Alternately, the loaded ruleset is retained in `/tmp/rules.debug`. If you are familiar with PF ruleset syntax, you can edit that to fix your connectivity issue and reload those rules like so:

```
# pfctl -f /tmp/rules.debug
```

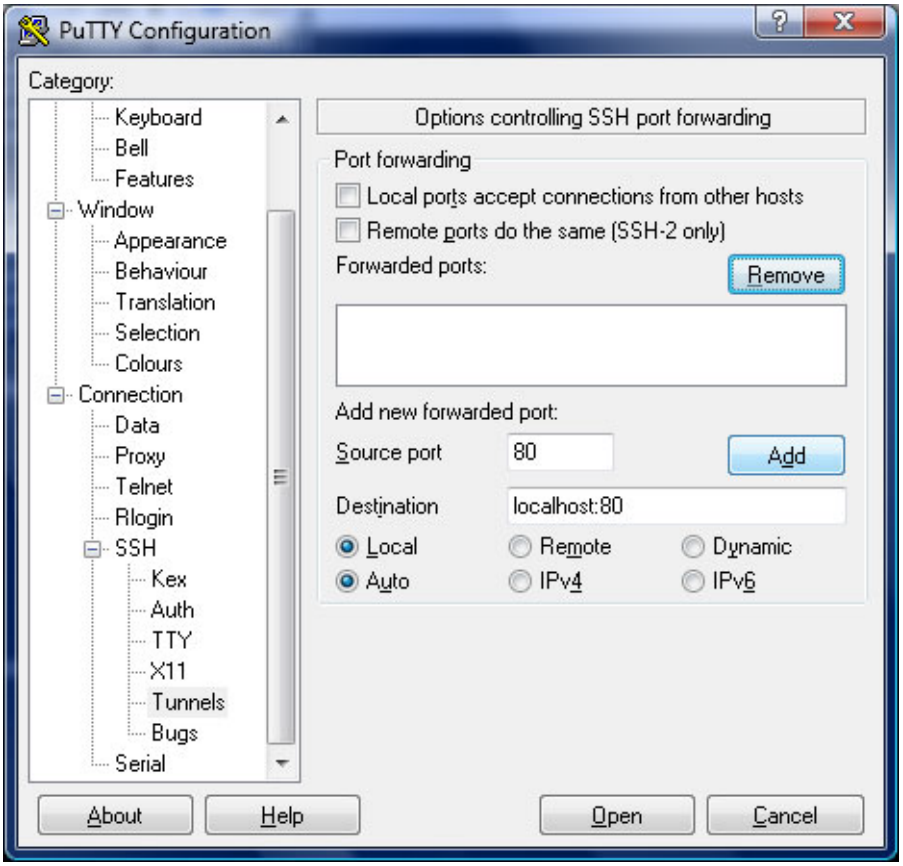
After getting back into the WebGUI with that temporary fix, do whatever work you need to do in the WebGUI to make the fix permanent. When you save the rules in the WebGUI, that temporary ruleset will be overwritten.

# Remotely Circumvent Firewall Lockout with SSH Tunneling

If you blocked access to the WebGUI remotely, but you still have access with SSH, then there is a relatively easy way to get in: SSH Tunneling.

If the WebGUI is on port 80, set your client to forward local port 80 (or 8080, or whatever) to remote port "localhost:80", then point your browser to **`http://localhost:80`** or whichever local port you chose. If your WebGUI is on another port, use that instead. If you are using HTTPS you will still need to use HTTPS to access the WebGUI this manner.

**Figure 5.16. Setting up a port 80 SSH Tunnel in PuTTY**



Fill out the options as shown in Figure 5.16, “Setting up a port 80 SSH Tunnel in PuTTY”, then click Add. Once you connect and enter your username/password, you can access the WebGUI using your redirected local port.

## Locked Out Due to Squid Configuration Error

If you accidentally configure Squid to use the same port as the WebGUI, and then cannot get back in to fix the configuration, you may need to fix it using the following procedure.

- Connect to the pfSense system console with SSH or physical access
- Start a shell, option 8 from the console.
- Terminate the squid process like so:

  ```
  # /usr/local/etc/rc.d/squid.sh stop
  ```

  If that doesn't work, try it this way:

  ```
  # killall -9 squid
  ```

  or

  ```
  # squid -k shutdown
  ```

Once the squid process is fully terminated, you should be able to regain access to the WebGUI. Be aware that you may need to work quickly, or repeat the shutdown command, as squid may be automatically restarted.

# NanoBSD-Specific Configuration

Several options specific to NanoBSD are available under Diagnostics → NanoBSD, which only shows up if you are running a NanoBSD-based image.

## Bootup Information

The Bootup Information section of the page shows you the NanoBSD Image size (512M, 1G, 2G, 4G) which is useful for determining which upgrade file to download, if needed. There is not a 1:1 correlation between the image size and the actual media size, but it would let you know the minimum size of the media.

The page also shows you the current Bootup Slice, such as *ad0s1*. If you want to switch to the other unused slice (to return to a previous firmware, escape some failed manual editing issues, etc), you may do so by pressing the Switch Slice button.

## Media Read/Write Status

On NanoBSD, the disk is kept read-only except during times when the system needs to write, typically when updating the config file or rewriting a package or service's configuration. If you need to make a manual change to a file on the system, you can do so by switching the disk to read/write mode by pressing Switch to Read/Write. Then Switch to Read-Only once you have completed your changes.

### Note

Be aware that because other processes on the system will attempt to switch between read/write and read-only modes, you may find that the system has automatically returned to a Read-Only state.

You can also perform this read/write and read-only switching from the shell, by using **/etc/rc.conf_mount_rw** and **/etc/rc.conf_mount_ro** respectively.

Should you find yourself in a situation where you do not wish the system disk to be kept read-only, you may check the box to Keep media mounted read/write at all times and then press Save. This will never allow the system to switch the disk back to read-only mode. This would be mostly useful for developers or those running NanoBSD on a hard drive instead of flash media.

## Duplicate Bootup Slice

Should you find yourself in a situation where the alternate/unused slice is corrupt or in some way broken, this option will let you duplicate the current working bootup slice over to the alternate slice. You can also use this after a major upgrade or change to ensure someone does not accidentally boot the old/incorrect slice and cause problems. There is typically only one entry in the Destination Slice drop down menu, and it would look something like **`ad0s1 -> ad0s2`**. Make sure that is the correct source/destination slice and then press Duplicate Slice.

## Upgrade Log

NanoBSD keeps a copy of the upgrade log on the config slice. Should a problem happen during the upgrade, you can click the View Upgrade Log button to see the log and determine what went wrong.

# Final Configuration Thoughts

There are millions of ways to configure a pfSense system, and thus it is impossible to cover all aspects of each configuration and troubleshooting in this book. This chapter provided an overview of some of the general configuration options. The coming chapters go into detail on individual capabilities of the software. As we mentioned at the end of the introductory chapter, there are several other avenues for getting help. If you have tried all of the suggestions here and you still aren't able to make pfSense perform as you expect, there are forums, IRC, mailing lists, Google searches, and Commercial Support. You are free to take the DIY approach, or if you would like professionals to take care of the configuration for you, the Commercial Support team is more than capable. For links to the online support mediums, refer to the section called “Getting Help”.

# Chapter 6. Interface Types and Configuration

Many different types of network interfaces can be used with pfSense, either using physical interfaces directly or by layering other protocols on top such as PPP or VLANs. In pfSense 1.2.3 this was primarily limited to using the interfaces themselves, VLANs, or PPPoE/PPTP. In pfSense 2.1, many new interface types are supported. Most of these were supported in pfSense 2.0.x, with the IPv6 types being the major addition for pfSense 2.1.

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). You can use various combinations of options to use the interfaces themselves, or use multiple networks and protocols on a single interface, or bind multiple interfaces together into a larger capacity or redundant virtual interface. All of the currently-defined interfaces are listed under Interfaces → (assign). By default, this is only the physical interfaces, but using the other tabs under Interfaces → (assign) you can create virtual interfaces and then assign them.

pfSense 1.2.3 limited you to only DHCP or static on LAN and OPT interfaces, but with 2.0 and above, you can use any interface type as any interface on pfSense. All interfaces are treated equally; You can rename the WAN and LAN interfaces to other names and use them in other ways.

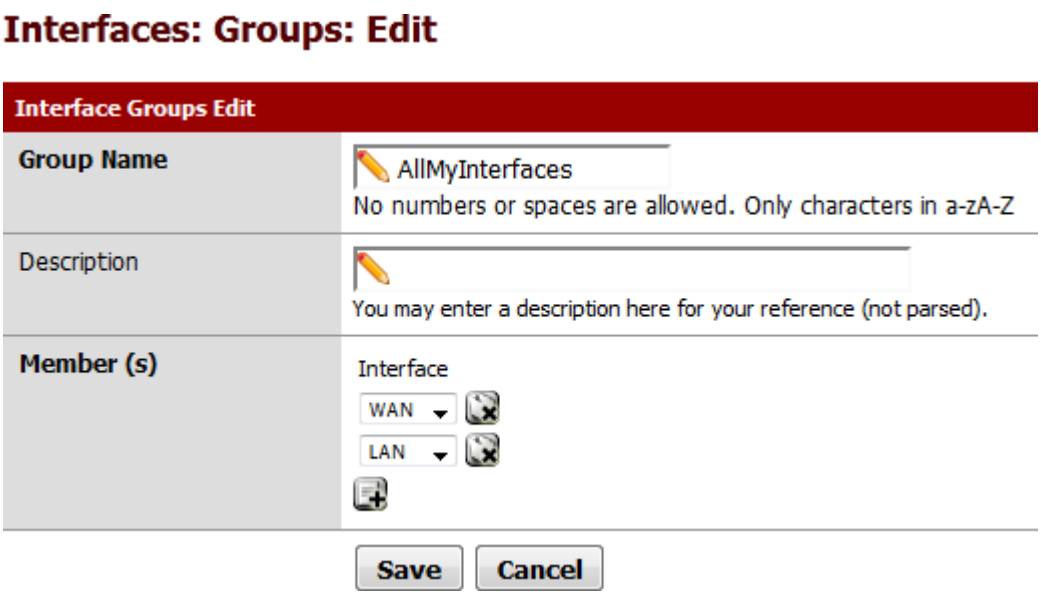
In this section, the various types of interfaces that can be created, assigned, and managed will be covered.

## 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 PPTP and OpenVPN work. There are multiple interfaces in the underlying OS, but the rules for all of them are managed on a single tab for each VPN type. If there will be many interfaces of a similar function that need practically identical rules, an interface group may be created to add rules to all of the interfaces at the same time. The interfaces can still have their own individual rules, which are processed before the group rules.

To create an interface group, navigate to Interfaces → (assign), and go to the Interface Groups tab. Click to create a new group. Then, enter a Group Name. The Group Name may only contain upper and lowercase letters, no numbers, spaces, or special characters. To add Member interfaces to the group, click and then select the Interface from the drop-down. To remove an interface from the group, click . When finished editing the group, click Save.

**Figure 6.1. Add Interface Group**




Interface groups get their own tab under Firewall → Rules, where their rules are managed. For more information on using Interface Group firewall rules, see .

**Figure 6.2. Interface Group Firewall Rules Tab**


# Wireless

The Wireless tab under Interfaces → (assign) is used to create 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 here, then assigned on the Interface assignments tab. In-depth information on this feature can be found in Chapter 23, *Wireless*.

# VLANs

VLAN tagged interfaces, or 802.1Q tagged interfaces, are managed on the VLANs tab under Interfaces → (assign). These allow the system to address traffic tagged by an 802.1Q capable switch separately as if each tag were its own interface. A VLAN is created here, then assigned on the Interface assignments tab. In-depth information on this feature can be found in Chapter 14, *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 the section called “pfSense QinQ Configuration”.

# PPPs

There are four types of PPP interfaces in pfSense 2.0 and above: Normal PPP for 3G/4G and modem devices, PPPoE for DSL or similar connections, and PPTP and L2TP for certain specific ISPs that require it for authentication in some regions. 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)

Aside from setting advanced options, one reason for editing the PPP interfaces here is to active Multi-Link PPP (MLPPP) with supported providers. This allows you to bond 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. The main downside to MLPPP is the difficulty of monitoring the link status for the individual lines.

Activating MLPPP is done by simply selecting more than one interface when editing a PPP type entry. Selecting multiple entries can vary by OS and browser, but most commonly it is performed by holding the **Ctrl** key while clicking on the interface names. When configuring MLPPP, be aware that the ISP must support MLPPP, all links need to be connected to the same ISP, and the same credentials must be valid for all links to be used concurrently. It also works best when the circuits are all of the same capacity, but can work with different speeds in some cases. If you are unsure if your ISP supports MLPPP, check with them before ordering an additional circuit for this purpose. Even if the provider does not support MLPPP it may still be possible to use the circuits separately with a traditional multi-WAN setup.

This topic is also discussed in Chapter 15, *Multiple WAN Connections*.

## PPP (Point-to-Point Protocol) Interface Types

When a PPP (Point-to-Point Protocol) entry is added or edited at Interfaces → (assign) on the PPPs tab, the first choice there is the Link Type. From there you can select one of the following types and configure options specific to that type.

### PPP (3G/4G, Modem)

The PPP link type is used for talking to a modem over a serial device. This can be anything from an old hardware modem for dial-up access to a USB 3G/4G dongle for accessing a cellular network. 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. You can click on a specific entry to select it for use. After selecting the interface, you may optionally enter a Description for the PPP entry.

If a 3G/4G network is in use, you may use the Service Provider options to pre-fill the remaining information on the page. First, select a Country, such as **`United States`**. The Provider list will appear with known cellular providers in that country. Select a Provider from the list, such as **`T-Mobile`**, and then the Plan list will become filled with items to select. When a Plan is chosen, the remaining fields will be filled in as needed with known values for that Provider and Plan.

The options can be configured manually if other values are needed, or for using a provider that is not listed. The Username and Password fields are the credentials used for the PPP login. The Phone Number is the ISP's number to dial to gain access. For dial-up this is probably a traditional phone number, but for 3G/4G this tends to be a number such as **`*99#`** or **`#777`**. The Access Point Name (APN) option is required by some ISPs to identify the service to which you're connecting. Some providers use this to distinguish between consumer and business plans, or legacy networks.

### PPPoE (Point-to-Point Protocol over Ethernet)

Most commonly found on DSL networks, PPPoE is a popular method of authenticating and gaining access to an ISP network. Upon selecting the PPPoE link type, the Link Interface(s) list is populated

with 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 at least one interface to use for this link. After selecting the interface, you may optionally enter a Description for the PPPoE entry.

You must at least fill in the fields for Username and Password. These will be provided by your ISP, and the username is typically in the form of an e-mail address, such as **`mycompany@ispexample.com`**. The Service Name name may be required by some ISPs, but is often left blank. If you are in doubt, leave it blank or contact your ISP and ask if it is necessary. Certain providers require that a value of **`NULL`** be sent for the service name. If that behavior is required by your ISP, check Configure a NULL Service name.

### 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. Upon selecting the PPTP link type, the Link Interface(s) list is populated with network interfaces that can be used for PPTP. These are typically physical interfaces but it can also work over some other interface types such as VLANs. Select at least one interface to use for this link. After selecting the interface, you may optionally enter a Description for the PPTP entry.

You must at least fill in the fields for Username and Password. These will be provided by your ISP.

### 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. You may refer to the previous section for configuration information.

## Advanced PPP Options

All PPP types have some advanced options in common that can be edited in their entries here. In most cases these settings need not be altered, but they are here if non-default settings are desired. When editing a PPP entry, these settings are accessed by clicking Show advanced options.

### 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. The Enable Dial-on-Demand mode will delay this connection attempt. When set, it will wait until a packet attempts to leave the firewall via this interface, and then it will connect. Once connected, it will not automatically disconnect.

### Idle Timeout

As mentioned previously, a PPP connection will be held open indefinitely by default. If you enter a value for Idle Timeout, specified in seconds, then the link will be monitored for activity. If there is no traffic on the link for that amount of time, the link will be disconnected. If Enable Dial-on-Demand mode has also been set, it will return to dial-on-demand mode.

#### Note

Be aware that pfSense will perform gateway monitoring by default which will generate one ICMP ping per second on the interface. If setting an Idle Timeout value isn't causing a disconnect when expected, the cause is likely the gateway monitoring traffic. 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. You almost always want to leave it 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 many setups 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 for some reason you do not want this behavior, 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 you are using MLPPP and need to disable this feature, check Disable shortseq.

### Address Control Field Compression (AFCComp)

Address and control field compression. This option only applies to asynchronous link types. It saves two bytes per frame. If you need to disable this, check Disable acfcomp.

### Protocol Field Compression (ProtoComp)

Protocol field compression. This option saves one byte per frame for most frames. If you need to disable this, check Disable protocomp.

## GRE (Generic Routing Encapsulation)

Generic Routing Encapsulation (GRE) is a method of tunneling traffic between two routers 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. This is the case with PPTP, which employs GRE to tunnel the traffic for the VPN after establishing an encrypted channel. IPsec, when in transport mode, can also use GRE for tunneling encrypted traffic. The GRE protocol was originally designed by Cisco, and it is the default tunneling mode on many of their devices.

The Parent interface of the GRE tunnel is the interface upon which the GRE tunnel will terminate. Often this will be WAN or a WAN-type connection.

The GRE remote address is the address of the remote peer. This is the address where the GRE packets will be sent, so it would be the routable address for the other end of the tunnel.

The GRE tunnel local address is the internal address for this end of the tunnel. The traffic inside of the tunnel will be sourced from this address, and tunneled remote traffic would be send to this address from the far side.

The GRE tunnel remote address is the address used inside the tunnel to reach the other end. Traffic going to the other end of the tunnel would be routed to this address.

If Mobile tunnel is selected, the tunneled traffic will be sent encoded in the IP protocol Mobile (55) rather than using GRE (47).

The GRE device needs a route to the destination that is less specific than the one over the tunnel to operate correctly. Basically, there needs to be a route to the remote endpoint that does not run over the tunnel, as this would be a loop and the traffic could never reach its destination. The Route search type option controls this behavior, and can be used if the remote endpoint as specified above in the GRE remote address is not the actual remote system receiving the GRE traffic.

By default, Web Cache Communication Protocol (WCCP) version 1 is used on the GRE tunnel. If you wish to use WCCP version 2 instead, check WCCP version.

Lastly, the Description field allows you to provide a short description of this GRE tunnel for documentation purposes.

# GIF (Generic tunnel InterFace)

GIF, short for Generic Tunneling Interface, is similar to GRE in that it is a means to tunnel traffic between two hosts. However, 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 [http://www.tunnelbroker.net/] and SixXS [http://www.sixxs.net/] in places where there is not yet any native IPv6 connectivity. See the section called “Connecting with a Tunnel Broker Service” for more on 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, using IPsec in transport mode, you can use a GIF tunnel between the endpoints to bridge layer 2 between two locations. Though we don't recommend the practice, there are some rare circumstances that require such a configuration.

As with GRE, the Parent interface of the GIF tunnel is the interface upon which the GIF tunnel will terminate. Often this will be WAN or a WAN-type connection.

The GIF remote address is the address of the remote peer, to which the GIF traffic will be sent. 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.

The GIF tunnel local address is the near side internal to the tunnel. For example, when tunneling IPv6-in-IPv4 via Hurricane Electric, they refer to this as the Client IPv6 Address.

The GIF tunnel remote address is the far side internal to the tunnel. For example, when tunneling IPv6-in-IPv4 via Hurricane Electric, they refer to this as the Server IPv6 Address. The CIDR drop-down is used to select the subnet size for the tunnel addresses. In this example it would be **64**.

The Route caching option controls whether or not the route to the remote endpoint is cached. If your 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.

The ECN friendly behavior option controls whether or not Explicit Congestion Notification (ECN)-friendly practice of copying the TOS bit into/out of the tunnel traffic is done. By default the TOS bit on the packets is cleared or set 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.

Lastly, the Description field allows you to provide a short description of this GIF tunnel for documentation purposes.

## Note

In most cases you will want to assign this GIF interface under Interfaces → (assign). When doing so, be sure to set the interface up as a Static IP or IPv6 interface, and use

the same IP address and subnet mask/prefix length as in the GIF tunnel local address. The GIF tunnel remote address should then be added as a gateway.

# 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, may be found in Chapter 13, *Bridging*.

# LAGG (Link Aggregation)

On FreeBSD, and thus pfSense, a `lagg(4)` interface (LAGG) is used for link aggregation. In other words, using multiple interfaces together as one. There are several ways this can work, either for gaining extra bandwidth, redundancy, or some combination of the two.

A LAGG interface can be made from selecting multiple unused interfaces. The Parent interface selection box on the LAGG editing screen is where this is done. Selecting multiple entries can vary by OS and browser, but most commonly it is performed by holding the **Ctrl** key while clicking on the interface names. An interface may only be added to a LAGG group if it is not assigned.

After creating a LAGG interface, you assign the lagg interface as any other under Interfaces → (assign) and give it an IP address, or you can build other things on top of it such as VLANs and assign those.

Due to a limitation in FreeBSD, `lagg(4)` does not support `altq(4)` so it is not yet possible to use the traffic shaper on LAGG or LAGG-based interfaces. As a workaround, you may use Limiters to control bandwidth usage on LAGG interfaces.

There are currently six different operating modes for LAGG interfaces: Failover, FEC, LACP, Load Balance, Round Robin, and None.

## Failover LAGG Protocol

When using the Failover LAGG protocol traffic will only be sent on the primary interfaces of the group, and if it 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.

## FEC LAGG Protocol

The FEC mode supports Cisco EtherChannel. The switch in use must support EtherChannel and have it configured accordingly. This is a static setup and does not negotiate aggregation with the peer or exchange frames to monitor the links. LACP should be used instead where it is supported.

## LACP LAGG Protocol

The most commonly used LAGG protocol is LACP. This mode supports the IEEE 802.3ad Link Aggregation Control Protocol (LACP) and the Marker Protocol. In LACP mode, negotiation is performed with the switch — which must also support LACP — to form a group of ports that are all active at the same time. This is knowns as a Link Aggregation Group, or LAG. The speed and MTU of each port in a LAG must be identical and the ports must also run at full-duplex. If link is lost to a port on the LAG, it continues to function but at reduced capacity. In this way, an LACP LAGG bundle can gain you both redundancy and increased bandwidth.

Traffic will be balanced between all ports on the LAG, however for communication between two single hosts it will only use up to 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 for most traffic patterns and is not relevant for failover.

In addition to configuring this option on pfSense, you will also need to configure your switch to enable LACP on these ports or to bundle them into a LAG group there. Both sides must agree on the configuration in order for it to work properly.

### Load Balance LAGG Protocol

The Load Balance mode will accept inbound traffic on any port of the LAGG group. It is a static setup that does not monitor the link state or do any negotiation with the switch. Outbound traffic is load balanced based on a hash computed using several factors, such as the source and destination IP/MAC and VLAN tag.

### Round Robin LAGG Protocol

The Round Robin mode will accept inbound traffic on any port of the LAGG group, and send outbound traffic using a round robin scheduling algorithm. Typically this means that traffic will be send out in sequence, using each interface in the group in turn.

### None LAGG Protocol

The None mode will disable traffic on the LAGG interface without actually disabling the actual interface. The OS will still believe the interface is up and usable, but no traffic will be sent or received on the group.

## OpenVPN

After they have been created, OpenVPN interfaces may be assigned under Interfaces → (assign). When this is done, the interface should be enabled under its Interfaces → OPTx page and set to an IP type of None for both IPv4 and IPv6. Assigning an OpenVPN interface will let you create interface-specific rules, and also use the OpenVPN interface elsewhere in the GUI that requires an assigned interface. If the OpenVPN interface being assigned is a client, 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.

# Interface Configuration

Once an interface has been assigned under Interfaces → (assign), it is allocated a default name such as OPT1, OPT2, etc. The first two interfaces are named WAN and LAN for historical reasons, but you may now rename at will. These OPTx names appear under the Interfaces menu, such as Interfaces → OPT1. Selecting the menu option for the interface will take you to that interface's configuration page.

If you have never used this interface before, you'll be greeted by a page containing only a single option — Enable Interface. By checking Enable Interface, the remainder of the options will appear.

## Description

The interface can be renamed by entering a new name into the Description box. 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. These interface names may only contain letters, numbers and the only special character that is allowed is an underscore ("_"). This makes it much easier to remember not only what an interface is for, but also to identify an interface for adding firewall rules or choosing other per-interface functionality.

## MAC address

You may change the MAC address of an interface should you need to spoof the MAC Address of a previous piece of equipment. Generally this should be avoided, as 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. In some cases it can be desirable to "clone" or "spoof" the MAC address of a previous piece of equipment. This 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 you may need to register the MAC with the ISP if and when it changes.

One downside to spoofing the MAC is that, unless the old piece of equipment is permanently retired, you run the risk of later having a MAC address conflict on your network, which can lead to connectivity problems. Also ARP cache problems tend to be very temporary, resolving automatically within minutes or by power cycling other equipment.

Should the old MAC address need to be restored, this box must be cleared out and then the firewall must be rebooted.

## MTU (Maximum Transmission Unit)

The Maximum Transmission Unit (MTU) size field can typically be left blank, but can be changed if desired. Some situations may call for a lower MTU to ensure packets are sized appropriately for your 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.

## MSS (Maximum Segment Size)

Similar to the MTU field, the MSS field will "clamp" 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 Operating System 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.

## Block Private Networks

If you select Block private networks, pfSense will insert a rule automatically that will prevent 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

If the Block bogon networks option is checked, pfSense will periodically download and block traffic from a list of unallocated and reserved networks. 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 networks should never be in active use on a network, especially one facing the Internet, so it's a good thing to use on WAN type interfaces. For IPv6, the list is still quite large, containing large 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 you will want to configure an IP address. For IPv4 connections, you can choose from: Static, DHCP, PPP, PPPoE, PPTP, and L2TP. These options are selected using the IPv4 Configuration Type selector.

# None

Setting the IPv4 Configuration Type to None will disable IPv4 on the interface. This is useful if the interface has no IPv4, or if the IP address on the interface is being managed in some other way, such as for an OpenVPN interface.

# Static IPv4

Selecting Static IPv4 will allow you to manually set the IP address for the interface to use. Using this option enables three additional fields on the interface configuration screen: IPv4 address, a CIDR subnet mask selector, and a Gateway field.

Enter the IPv4 address for the interface into the IPv4 address box, and choose the subnet mask from the CIDR drop-down after the address box.

If this is a WAN type interface, you should select a Gateway or add one if one does not already exist. To pick one that already exists, click and selected from the drop-down list. If you click add a new one, a form will appear to add the gateway. If the form field does not appear, try a different web browser. Historically, Internet Explorer has had issues with some of our JavaScript and AJAX driven forms, this one especially, but it should work fine with current IE versions in 2.1 and newer.

To add a gateway, after clicking add a new one, fill in the details requested on the new form. If this is the only WAN or will be a new default WAN, check Default gateway. The Gateway Name is used to refer to the gateway internally, as well as in places like Gateway Groups, the Quality Graphs, and elsewhere. The Gateway IPv4 field is where you enter the actual gateway IP address. This address must be inside of the same subnet as the Static IPv4 address. The Description box allows you to enter a bit of text to indicate the purpose of the gateway. When finished, click Save Gateway.

## Note

Selecting a Gateway from the drop-down list, or adding a new gateway and selecting it, will make pfSense 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. You may still use gateways on those interfaces for the purpose of static routes without selecting a Gateway here on the interfaces screen.

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.

# DHCP

Choosing DHCP from the list will cause pfSense to attempt automatic IPv4 configuration of this interface via DHCP. This option also activates three additional fields on the page: Hostname, Alias IPv4 address, and a CIDR drop-down for Alias IPv4 address. Under most circumstances these additional fields may simply be left blank.

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.

The value entered in the Alias IPv4 address field is used as a fixed alias IPv4 address by the DHCP client. 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's management IP address. With a static IPv4 address you could simply add an IP alias type VIP, but since that is not available on DHCP, this option allows one to be configured.

The Reject Leases From box allows you to put in an IPv4 address for a DHCP server that should be ignored. For example, if you have a cable modem that hands out private IPs when the cable sync has

been lost, you can enter the modem's private IP here, e.g. *192.168.100.1*, and your firewall will never pick up the private IP and attempt to use it.

## PPP Types

The various PPP-based connection types such as PPP, PPPoE, PPTP, and L2TP were all covered in detail earlier in this chapter (the section called “PPPs”). When you select them here on the interfaces screen you can set or change their basic options 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, the exact method you will need to use depends on the network to which you are connected and how the ISP has deployed IPv6 on that network. For more information on IPv6, including a basic introduction, see the section called “IPv6”.

## None

Setting the IPv6 Configuration Type to None will disable IPv6 on the interface. This is useful if the interface has no IPv4, or if the IP address on the interface is being managed in some other way, such as for an OpenVPN interface.

## Static IPv6

Selecting Static IPv6 will allow you to manually set the IPv6 address for the interface to use. Using this option enables three additional fields on the interface configuration screen: IPv6 address, a Prefix Length selector, and a Gateway field.

Enter the IPv6 address for the interface into the IPv6 address box, and choose the prefix length from the drop-down list after the address box.

If this is a WAN type interface, you should select a Gateway or add one if one does not already exist. To pick one that already exists, click and selected from the drop-down list. If you click add a new one, a form will appear to add the gateway. If the form field does not appear, try a different web browser. Historically, Internet Explorer has had issues with some of our JavaScript and AJAX driven forms, this one especially, but it should work fine with current IE versions in 2.1 and newer.

To add a gateway, after clicking add a new one, fill in the details requested on the new form. If this is the only IPv6 WAN or will be a new default IPv6 WAN, check Default v6 gateway. The Gateway Name IPv6 is used to refer to the gateway internally, as well as in places like Gateway Groups, the Quality Graphs, and elsewhere. The Gateway IPv6 field is where you enter the actual gateway IP address. This address must be inside of the same subnet as the Static IPv6 address. The Description box allows you to enter a bit of text to indicate the purpose of the gateway. When finished, click Save Gateway.

### Note

Selecting a Gateway from the drop-down list, or adding a new gateway and selecting it, will make pfSense treat that interface as a WAN type interface. This is not desirable for internal-facing interfaces, such as LAN or a DMZ. You may still use gateways on those interfaces for the purpose of static routes without selecting a Gateway here on the interfaces screen.

The default IPv6 and IPv4 gateways work independently of one another. The two need not be on the same circuit. Changing the default IPv6 gateway has no effect on the IPv4 gateway, and vice versa.

# DHCP6

Choosing DHCP6 from the list will cause 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 still 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 the section called “Router Advertisements”.

When DHCPv6 is active, another field is also available: DHCPv6 Prefix Delegation size. If your ISP is providing you with a routed IPv6 network via prefix delegation, they will tell you 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 the section called “DHCP6 Prefix Delegation”. To use this delegation, you should set another internal interface's IPv6 Configuration Type to be Track Interface (the section called “Track Interface”) so that it can use the addresses delegated by the upstream DHCPv6 server.

# SLAAC

Choosing Stateless address autoconfiguration, or SLAAC, as the IPv6 type will make 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 the section called “Router Advertisements”.

# 6RD Tunnel

6RD is am IPv6 tunneling technology employed by some ISPs to quickly enable IPv6 support for their networks, passing IPv6 traffic inside specially crafted IPv4 packets between the user's router and the ISP's relay. It is related to 6to4 but is intended to be used within the ISP's network, using the ISP's IPv6 addresses for client traffic. To use 6RD, your ISP should have supplied you with three pieces of information: The 6RD prefix, the 6RD Border Relay, and the 6RD IPv4 Prefix length.

In the 6RD prefix box, enter the 6RD IPv6 prefix assigned by your ISP, such as 2001:db8::/32.

The 6RD Border Relay is the IPv4 address of your ISP's 6RD relay.

The 6RD IPv4 Prefix length controls how much of the end user's 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's allocation is all 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 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 just at the user's 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 is 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.

After selecting Track Interface, the IPv6 Interface option appears which lists 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 be receiving the delegated subnet information from the ISP.

If the ISP has delegated more than one prefix via DHCPv6, the IPv6 Prefix ID controls which of the delegated subnets will be used on this interface. This value is specified in hexadecimal. If you are unsure what to put here, leave it blank or contact your ISP.

For more information on how prefix delegation works, see the section called “DHCP6 Prefix Delegation”.

# Chapter 7. User Management and Authentication

In pfSense 2.0, the User Manager was introduced to allow adding multiple users to the GUI. These users 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.

When this was changed from the old pfSense 1.2.3 style, it made a custom login page possible, and also made it possible to log the current user out via System → Logout.

# 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. Users from RADIUS or LDAP still need to have 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. |
| Captive Portal | Supports users in the User Manager, and RADIUS users via settings in the Captive Portal page. |
| PPTP | Supports users in the PPTP settings, and via RADIUS in the PPTP settings. |
| 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. |

# User Management

The User Manager is located at System → User Manager. From there you can maintain users, groups, servers, and change settings that govern the behavior of the User Manager.

## Privileges

Managing permissions for users and groups is done similarly, so we'll cover it here rather than duplicating the effort. Whether you manage a user or group, the entry must be created and saved first before you can add permissions to the account or group. To add permissions, when editing the existing user or group, click in the Assigned Privileges or Effective Privileges section.

A list of permissions is shown on the screen, containing all possible permissions available. By default they are all unselected. You can add permissions one by one, or by multi-select. If there are other permissions already present on the user or group, they are hidden from this list so they can't be added twice.

Selecting a permission will show a short description of what it does in the Description area under the permission list. Most of the permissions are pretty 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 just the dashboard page and all of its associated functions (widgets, graphs, etc.) |
| WebCfg - System: User Password Manager Page | If the user has access to just 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. |

On pfSense 2.1, the system menu displayed on the page only shows menu entries to which the user has access. On pfSense 2.0.x, all menu items were shown but the user would be denied access to the ones they could not load.

# Adding/Editing Users

The Users tab under System → User Manager is where individual users are managed. To add a new user, click , to edit an existing user, click .

Before you can add permissions to a user, it must first be created, so the first step is always to add the user. If you will have multiple users that need the same permissions, it is easier to add a group and then add users to the group.

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

The Disabled checkbox controls whether this user will be active. If you need to deactivate this account, you can check this box.

The Username is required, and must be 16 characters or less and may only contain letters, numbers, and a period, hyphen, or underscore.

The Password fields are also required. These passwords are stored in the pfSense configuration as hashes. Ensure the two fields match to confirm the password.

The optional Full Name field can be used to enter a user's name or a description for account.

An Expiration Date may also be defined if you wish to deactivate the user automatically when that date has been reached. The date should be entered in *MM/DD/YYYY* format.

If you have already defined groups and wish to add a user to them, then you can use the Group Memberships control to define which groups of which the user will be a member. To add a group for this user, find it in the list, 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.

The Effective Privileges part of the user editor only appears if you are editing an existing user. It does not show up when adding a user. See the section called “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.

The behavior of the Certificate portion of the page changes depending on whether you are adding a user or editing a user. 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 the section called

"Create an Internal Certificate". If you are editing a user, this section of the page instead becomes a list of user certificates. From here, you can click to add a certificate to the user. The settings on that page are identical to the section called "Create an Internal Certificate" except even more of the data is pre-filled with the user's name. If the certificate already exists, you can select **Choose an Existing Certificate** and then pick an Existing Certificate from the list.

The Authorized keys part of the user config allows you to paste in a user's SSH public key for shell or other SSH access. To add a key, check Click to paste an authorized key and then paste in the key data.

The IPsec Pre-Shared Key field can be used for a non-xauth mobile IPsec setup. If an IPsec Pre-Shared Key is entered here, the username is used as the identifier. You can also see the PSK under VPN → IPsec on the Pre-Shared Keys tab. If you will only be using mobile IPsec with xauth, this field can be left blank.

After saving the user, you can click on the user's row to edit the entry.

# 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, you could have a group for IPsec xauth users, or a group that can access the firewall's dashboard, a group of firewall administrators, or whatever else you can imagine using our permissions system.

As with users, in order to add permissions to a group, it must be created first, and then you must save and go back to edit the group.

Groups are managed under System → User Manager on the Groups tab. To add a new group from this screen, click . To edit an existing group, click .

When adding a group, the first thing to do is give it a Group name. The Group name 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.

Next, you can optionally give the group a Description for your reference, to better identify the purpose of the group in case the Group name is not sufficient.

If you have already defined users and wish to add them to this group, then you can use the Group Memberships control to define which users will be members. To add a user to this group, find it in the list, select it, and click to move it to the Members column. To remove a user from the group, select it from the Memberscolumn and click to move it to the Not Members column.

If you are editing an existing group, then the Assigned Privileges section will be present on the screen. This section allows you to add permissions to the group. See the section called "Privileges" earlier in this section for information on managing privileges.

# Settings

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

The Session Timeout settings 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). You can enter a value of *0* to disable session expiration, making the login sessions valid forever, but we do not recommend doing this as it's a potential security risk.

The Authentication Server selector lets you choose 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

group memberships should 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.

# Authentication Servers

Using the Servers tab under System → User Manager, RADIUS and LDAP servers may be defined as authentication sources. See the section called “Support Throughout pfSense” for information on where these servers may be used in pfSense currently. To add a new server from this screen, click . To edit an existing server, click .

# RADIUS

To add a RADIUS server to pfSense, first fill in the Descriptive name field with the name for this RADIUS server. This name will be used to identify the server throughout the pfSense GUI. Make sure that your RADIUS server has the firewall defined as a client before attempting to set the values on this page.

From the Type field, select **RADIUS**. The Radius Server Settings will be displayed.

The first of these settings, Hostname or IP address, is the address of the RADIUS server. This can be a fully qualified domain name, an IPv4 IP address, or an IPv6 IP address.

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

The Services offered selector allows you to choose which services are offered by this RADIUS server. You can choose **Authentication and Accounting**, just **Authentication**, or just **Accounting**. Changing this selection controls which RADIUS services to use for this server, and it controls which of the port values below will show on the page. **Authentication** will use this RADIUS server to authenticate users. **Accounting** will send RADIUS start/stop accounting packet data for login sessions.

If authentication is one of the services in use, then the Authentication port value setting will appear. The default RADIUS authentication port is *1812*, but if your server uses a different port, specify it here.

If accounting is one of the services in use, then the Accounting port value setting will appear. The default RADIUS accounting port is *1813*, but if your server uses a different port, specify it here.

The Authentication Timeout box 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 you are using an interactive two-factor authentication system, 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 you need to wait for an external action such as a phone call, SMS message, etc.

Click Save and then this server will be ready to use.

# LDAP

To add an LDAP server to pfSense, first fill in the Descriptive name field with the name for this LDAP server. This name will be used to identify the server throughout the pfSense GUI.

From the Type field, select **LDAP**.

The first of these settings, Hostname or IP address, is the address of the LDAP server. This can be a fully qualified domain name, an IPv4 IP address, or an IPv6 IP address.

The Port value specifies which port on the LDAP server is listening for LDAP queries. The default TCP port is *389*, and for SSL it is *636*. 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 your LDAP server is listening on the correct port with the correct mode.

Transport 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 your 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 your LDAP server is not on a local network segment. 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.

If you chose **`SSL - Encrypted`** for the Transport, then you must select a Peer Certificate Authority to validate the certificate of the LDAP server. If you are connecting to an Active Directory server, you must make sure the selected CA matches the CA configured on your LDAP server, otherwise problems will arise. See the section called “Certificate Authority Management” for more information on creating or importing CAs.

## Note

When connecting to LDAP with SSL, you must use a Hostname for the LDAP server, not an IP address. The hostname given for the server is also used to verify the server certificate. The common name of the server certificate must be its hostname, and that hostname must resolve to the LDAP server's IP address, 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 if they do not match in your network infrastructure and they cannot be easily fixed.

The Protocol version selector allows you to choose which version of the LDAP protocol is employed by your LDAP server, either **`2`** or **`3`**.

Search scope determines where, and how deep, a search will go for a match. Under Level, you can choose between **`One Level`** or **`Entire Subtree`** to control how deep the search will go. The Base DN field controls where the search will start.

Authentication containers is a semicolon-separated list of potential account locations, or containers. These containers will be prepended to the search Base DN above or you can specify full container path here and leave the Base DN blank. If your LDAP server supports it, and your bind settings are correct, you can click the Select button to browse the LDAP server's containers and select on that one. Some examples of these containers would be:

- *`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.

The Extended Query box lets you specify an extra bit to query after the username, which allows you to specify a group membership to use as a filter. To set an Extended Query, check the box and fill in the value with something like *`CN=Groupname,OU=MyGroups,DC=example,DC=com`*.

The Bind credentials option 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 your 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.

The Initial Template selector will pre-fill 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 is the attribute used to identify a user's name, most commonly *cn* or *samAccountName*.

Group naming attribute is the attribute used to identify a group, such as *cn*.

Group member attribute is the attribute of a user that signifies it is the member of a group, such as *member*, *memberOf*, or *uniqueMember*.

Click Save and then this server will be ready to use.

# 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 your server setup may vary from the one used.

## RADIUS Server Example

This example was made against FreeRADIUS but doing the same for Windows Server would be identical. See the section called “RADIUS Authentication with Windows Server” for info on setting up a Windows Server for RADIUS.

This assumes you have already setup your RADIUS server to accept queries from this firewall as a client with a shared secret.

| | |
|---|---|
| Descriptive Name | *ExCoRADIUS* |
| Type | **Radius** |
| Hostname or IP Address | *172.17.0.1* |
| 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 | *2001:470:1f11:e1c::2:101* |
| Port | *389* |
| Transport | **TCP - Standard** |

| | |
|---|---|
| Protocol Version | **3** |
| Search Scope | **One Level**, *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 | *member* |

# Active Directory LDAP Example

In this example, pfSense is setup to connect to the company Active Directory structure in order to authenticate users.

| | |
|---|---|
| Descriptive Name | *ExCoAD* |
| Type | **LDAP** |
| Hostname or IP Address | *192.168.2.230* |
| Port | *389* |
| Transport | **TCP - Standard** |
| Protocol Version | **3** |
| Search Scope | **One Level**, *DC=domain,DC=local* |
| Authentication Containers | *CN=Users,DC=domain,DC=local* |
| 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 you import your AD structure's Certificate Authority under the Certificate Manager in pfSense, you can use SSL as well by selecting that option and choosing the appropriate CA from the Peer Certificate Authority drop down on this page.

# Troubleshooting

There is a page available to test these authentication servers at Diagnostics → Authentication. From that page, you can select your Authentication Server, enter a Username and Password, and then press the Test button. The firewall will attempt to authenticate that 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 you receive an error when testing the authentication, double check your server settings, make any necessary adjustments and then 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. Enter the full Distinguished Name (DN) for the bind user, such as *CN=binduser,CN=Users,DC=domain,DC=local*.

If you are unsure of the user's full DN, you can find it by navigating to the user in **ADSI Edit** found under Administrative Tools on the Windows Server.

# Active Directory Group Membership

Depending on how your Active Directory groups were made, the way you specify them 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 should be.

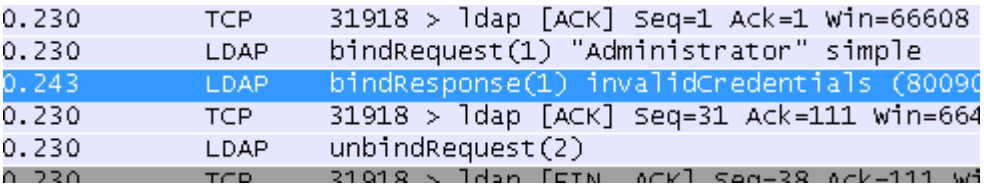
# Troubleshooting via Server Logs

Authentication failures should be logged by your 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 you're using an unencrypted method (RADIUS, LDAP without SSL), you may not see the actual password being used, but you can see enough of the protocol exchange to determine why a request is failing to complete. This is especially true when you load a capture up in Wireshark, which can interpret the responses for you, as seen in Figure 7.1, “Sample LDAP Failure Capture”. For more information on packet captures, see Chapter 30, *Packet Capturing*.

**Figure 7.1. Sample LDAP Failure Capture**




# Troubleshooting via LDAP Debugging

If SSL is being used with LDAP, capturing packets is not possible, so it makes debugging the connection much more difficult. We have a patch available that enables debug logging for LDAP, and the patch works with the System Patches package (http://doc.pfsense.org/index.php/System_Patches). Once the System Patches package has been installed, add a new patch with the following settings:

| | |
|---|---|
| URL | http://files.pfsense.org/jimp/patches/ldap-debug.patch |
| Path Strip | **1** |
| Base | **/** |

Ignore Whitespace **Checked**

Once applied, the WebGUI needs restarted. This can be done via reboot, or at the console or ssh run option 11 to restart the WebGUI. The patch enables debug logging for LDAP and also tells the WebGUI server process to write the error log for those debug messages. Once the WebGUI has restarted, try your LDAP query again and then check `/var/log/lighttpd-breakage.log` which should have sufficient detail to track down the cause of the problem. If you have trouble understanding the debug output, post in the forum for help or seek assistance from pfSense Commercial Support.

# Chapter 8. Certificate Management

# 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 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 you'll need 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 your PKI. The CA's certificate is used on the OpenVPN servers and clients to verify the authenticity of certificates used. The CA's certificate can be used to verify the signing on certificates, but not to sign certificates. Signing certificates requires the CA's private key. The privacy of the CA private key is what ensures the security of your PKI. Anyone with access to the CA private key can generate certificates to be used on your PKI, hence it must be kept secure. This key is never distributed to clients or servers.

Ensure you never copy more files to the clients than are needed, as this may result in the security of your PKI being compromised.

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 is it often suggested that you create a unique CA for each VPN that has a different level of security. For instance, if you have two mobile access VPNs with the same security access, using the same CA for those VPNs is OK. However if you have a VPN for users and a VPN for remote management, each with different restrictions, then you should use a unique CA for each VPN.

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, you must have the CA's private key imported into the GUI.

# Certificate Authority Management

Certificate Authorities are managed from System → Cert Manager, on the CAs tab. From this screen you can add, edit, export, or delete CAs.

## Create a new Certificate Authority

To create a new CA, go to System → Cert Manager on the CAs tab. From there, click the button to start the process of adding a CA.

First, enter a Descriptive name for the CA. This is used as a label for this CA throughout the GUI. How you proceed from here depends on what kind of CA you're trying to add. The Method drop-down selector has three options, each described below, Create an Internal Certificate Authority, Import an Existing Certificate Authority, and 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 you enter on this screen.

The Key length field chooses 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.

The Digest Algorithm drop-down has a list of possible digest algorithms that can be used for the certificate. Previously, this defaulted to only using SHA1, but the current recommendation is to use an algorithm stronger than SHA1 where possible. **SHA256** is a good choice.

The Lifetime field specifies the number of days for which the CA will be valid. How often you want to do this depends on your 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 just short of 10 years.

The Distinguished name section determines what personalized parameters will go into the CA. These are typically filled in with your organization's information, or in the case of an individual, your personal information. This information is mostly cosmetic, and used to verify the accuracy of the CA, and to distinguish one CA from another. The Country Code, State or Province, City, Organization, and Email Address should all be filled in with your information. The Common Name (CN) 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, you could call it *VPNCA* or *MyCA*. Although it is not considered an invalid CN, it is best to avoid using spaces in the CN.

When you have finished entering the information, press Save. If there are any errors, 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 you have an existing CA from an external source that you need to import, it can be done by selecting the Method of Import an Existing Certificate Authority. This can be useful in two ways. One, for CAs you have made yourself using another system, and two, for CAs made by others that you need to trust.

If you are trusting a CA from another source, you need only enter the Certificate data for the CA. 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-----
```

If you are importing your own CA, or a CA made for you that is capable of generating its own certificates and certificate revocation lists, you will also need to import the CA's private key. 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-----
```

If you imported the CA's private key, it is essential that you enter the Serial for next certificate value. 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 you will have problems later with certificate revocation. If you do not know what the next serial should be, attempt to estimate how many certificates have been made from the CA, and then set the number high enough that there should not be a collision.

## Importing a Chained or Nested Certificate Authority

If your CA has been signed by an intermediary and not directly by a root CA, you may need 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 let you 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. You can choose the higher-level CA to sign this CA's certificate using the Signing Certificate Authority drop-down. Only CAs with keys present will be shown, as this is required to properly sign this new CA. The remaining parameters for creating this CA are 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 button at the end of its row. The screen presented lets you edit the fields as if the CA were being imported. For information on the fields on this screen, see the section called “Import an Existing Certificate Authority”. In most cases the purpose of this screen would be to correct the CA's serial if needed using the Serial for next certificate entry, or to add a key to an imported CA so it can be used to create and sign certificates and CRLs. Once you have adjusted the settings you need, click Save.

## Export a Certificate Authority

From the list of CAs at System → Cert Manager on the CAs tab, you can also export a CA's certificate and/or private key. In most cases you would not want to export the private key for a CA, unless you are moving the CA to a new location, or making a backup. When using the CA for a VPN or most other purposes, you need only export the CA's certificate. If the CA's private key gets into the wrong hands, the other party could generate new certificates that would be considered valid against the CA. To export the CA's certificate, click the button on the *left*. To export the CA's private key, click the button on the *right*. To confirm you are exporting the proper file, hover your mouse over the button and a tooltip will display the action to be performed. The files will download with the CA's descriptive name as the file name, and 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. If it's being used by a VPN or other subsystem, it must be removed from there. Then, visit System → Cert Manager on the CAs tab. Find the CA in the list, click the button, and then click OK on the confirmation dialog. If you receive an error, 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 you can add, edit, export, or delete certificates.

## Create a new Certificate

To create a new certificate, go to System → Cert Manager on the Certificates tab. From there, click the button to start the process of adding a certificate.

First, enter a Descriptive name for the certificate. This is used as a label for this CA throughout the GUI. How you proceed from here depends on what kind of certificate you're trying to add. The Method drop-down selector has three options, each described below, Import an Existing Certificate, Create an Internal Certificate, and Create a Certificate Signing Request.

## Import an Existing Certificate

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

One must enter both the Certificate data and the Certificate's Private key data. To start, enter the Certificate data. 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-----
```

Next, you must to import the Private key data. 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 you include the entire block, including the beginning header and ending footer around the encoded data.

## Create an Internal Certificate

The most common Method used from here is to Create an Internal Certificate. This will make a new certificate using one of your existing certificate authorities.

First, 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.

The Key length field chooses 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.

The Digest Algorithm drop-down has a list of possible digest algorithms that can be used for the certificate. Previously, this defaulted to only using SHA1, but the current recommendation is to use an algorithm stronger than SHA1 where possible. **`SHA256`** is a good choice.

The Certificate Type field lets you set the purpose for this certificate. If the certificate will be used in a VPN server or HTTPS server, choose Server Certificate. This indicates inside the certificate that it may be used in a server role, and no other. If User Certificate is chosen, 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. Neither the Server Certificate nor User Certificate can be used to create additional certificates. If you want to create an intermediate CA, choose Certificate Authority. 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".

The Lifetime field specifies the number of days for which the certificate will be valid. How often you want to do this depends on your personal preferences and site policies. Changing the certificate frequently is more secure, but it is also a management headache as it would require reissuing a new

certificate when the current certificate expires. By default the GUI suggests using 3650 days, which is just short of 10 years.

The Distinguished name section determines what personalized parameters will go into the certificate. Most of these fields will be pre-populated with data from the CA. These are typically filled in with your organization's information, or in the case of an individual, your personal information. This information is mostly cosmetic, and used to verify the accuracy of the certificate, and to distinguish one certificate from another. The Country Code, State or Province, City, Organization, and Email Address should all be filled in with your information. The Common Name (CN) 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, you could call it *`VPNCert`*, *`user01`*, or *`vpnrouter.example.com`*. Although it is not considered an invalid CN, it is best to avoid using spaces in the CN.

If the certificate has multiple names that should be valid for the CN, such as two different hostnames, an additional IP address, a URL, or an e-mail address, these can be managed under Alternative Names. If you do not know what to enter for this field, it should probably be left blank. To add a new Alternative Name, click , and then a row with a Type and Value field will appear. The Type field must contain one of **`DNS`** (FQDN or Hostname), **`IP`** (IP address), **`URI`**, or **`email`**. the Value field must contain an appropriately formatted value based on the Type entered. If you decide you do not need a row, it can be removed by clicking the  at the end of the row.

When you have finished entering the information, press Save. If there are any errors, 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

A Certificate Signing Request will let you create a new request file that can be sent into a third party CA to be signed. This would be used if you wish to obtain a certificate from a trusted root certificate. To create one, select Certificate Signing Request from the Method drop-down. 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, you can also export a certificate and/or its private key. To export the certificate, click the  button on the *left*. To export the certificate's private key, click the  button in the *middle*. To export the certificate and its private key together in a PKCS#12 file, click the  button on the *right*. To confirm you are exporting the proper file, hover your mouse over the button and a tooltip will display the action to be performed. The files will download with the certificate's descriptive name 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. If it's being used by a VPN or other subsystem, it must be removed from there. Then, visit System → Cert Manager on the Certificates tab. Find the certificate in the list, click the  button, and then click OK on the confirmation dialog. If you receive an error, 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, depending on where the authentication for the VPN is being performed and whether or not the certificate already exists.

If there is no user authentication, or if the user authentication is being done on an external server (RADIUS, LDAP, etc) then you can make a user certificate just like any other certificate described

earlier. Ensure that you select User Certificate for the Certificate Type, and set the Common Name to be the user's username.

If the user authentication is being performed on pfSense, the user certificate can be made inside of the User Manager (System → User Manager) when creating the user. Inside there, add a new user, fill in the Username and Password, and in the User Certificates section, select Click to create a user certificate. This will show a simple form for creating a user certificate. Enter a short Descriptive Name, it can be the username or something such as *`Bob's Remote Access VPN Cert`*. Choose the proper Certificate authority for the VPN. The Key Length and Lifetime may also be adjusted if needed. When you save the user, a certificate will be generated for them.

To add a certificate to an existing user, edit the user, find the User Certificates and click . From there you can choose any of the other options available from the certificate creation process described in the section called “Create a new Certificate”, or you may also Choose an existing certificate to create an association between this user and a certificate that has already exists.

For more information on adding and managing users, see Chapter 7, *User Management and Authentication*.

# Certificate Revocation List Management

Certificate Revocation Lists (CRLs) are a part of the X.509 system that allow you to publish a 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 should also use a CRL, so it can verify connecting clients 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, you must have the private key of the CA imported into the GUI. If you manage the CA externally and do not have the CA's private key on the firewall, you can still import a CRL generated outside of the firewall.

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, you can create multiple CRLs for a single CA, but only one CRL may be chosen for a VPN instance. This could be used, for example, to prevent a specific certificate from connecting to one instance while allowing it to connect to another.

Certificate Revocation Lists are managed from System → Cert Manager, on the Certificate Revocation tab. From this screen you can add, edit, export, or delete CRL entries. The list will show all of your 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.

## Create a new Certificate Revocation List

To create a new CRL, find the row with the CA that the CRL will be created for, then click  at the end of the row.

On the next screen, for Method choose **`Create an Internal Certificate Revocation List`**.

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 CA's name and/or the CRL's purpose in this name.

From the Certificate Authority drop-down menu, ensure that the proper CA is selected.

In the Lifetime box, enter how long you wish the CRL to be valid. The default value is *`9999`* days, or almost 27 and a half years.

Now click Save and you will be returned 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, find the row with the CA that the CRL will be imported for, then click at the end of the row.

On the next screen, for Method choose **`Import an Existing Certificate Revocation List`**.

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 CA's name and/or the CRL's purpose in this name.

From the Certificate Authority drop-down menu, ensure that the proper CA is selected.

Next, you must to import 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 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 CRL. Make sure you include 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, you can also export a CRL. To export the CRL, click the button .The file will download with the CRL descriptive name as the file name, and the extension `.crl`.

# Delete a Certificate Revocation List

To remove a CRL, first it must be removed from active use. If it's being used by a VPN or other subsystem, it must be removed from there. Then, visit System → Cert Manager on the Certificate Revocation tab. Find the CRL in the list, click the button, and then click OK on the confirmation dialog. If you receive an error, follow the on-screen instructions to correct the problem and then try again.

# Revoke a Certificate

A CRL isn't very useful unless it has revoked certificates listed. A certificate is revoked by adding the certificate to a CRL, and do to this, edit an internal CRL by clicking. A screen will be presented that lists any currently revoked certificates, and a control to add new ones.

On this screen, there is a drop down list labeled Choose a Certificate to Revoke and this list contains all of the certificates known to the firewall for the CA used by this CRL. Select the certificate you wish to revoke from this list.

The Reason field allows you to indicate why a certificate is being revoked. This information doesn't affect the validity of the certificate it is merely informational in nature. You can select any of the values in the list if they apply, or leave it at the default.

Click Add and the certificate will be added to the CRL.

You can remove a certificate from the CRL from this screen as well. Find the certificate in the list and click the button to remove it from the CRL.

After adding or removing a certificate, the CRL will be re-written if it is currently in use by any OpenVPN instances so that the CRL changes will be immediately active.

## Updating an Imported Certificate Revocation List

To update an imported CRL, find it in the list and click the at the end of its row. You can then erase the pasted content in the CRL Data box and replace it with the contents of the new CRL, and press Save.

After updating the imported CRL, it will be re-written if it is currently in use by any OpenVPN instances so that the CRL changes will be immediately active.

# Import from EasyRSA

On pfSense 1.2.3, certificates could not be managed from the GUI so we had recommended creating and managing a certificate structure in EasyRSA. On pfSense 2.x, it's much easier to manage certificates in the GUI, and these will be backed up and restored in the config so it's also safer. So what to do if you upgrade? When you upgrade from 1.2.3 to 2.0 the upgrade process will import your existing CA certificate(s), and the certificates entered into the boxes for the OpenVPN clients/servers. It will not import your CA key or certificates for remote access clients because those were not stored in the config anywhere.. If you followed the old EasyRSA how-to these should still be in the old keys folder should be under `/root/easyrsa4pfsense/keys`.

If that folder is missing and you do not have a backup, then there is no way to generate new certificates from this CA. If you have these files backed up somewhere, locate the backup the files. Assuming the files are present, Login to the shell, then run:

```
# cat /root/easyrsa4pfsense/keys/ca.key
-----BEGIN RSA PRIVATE KEY-----
[...]
-----END RSA PRIVATE KEY-----
```

That will show you the existing CA key. Then from the GUI, go to System → Cert Manager, find the imported CA, and click the button to bring up its edit screen. Copy/paste the key (including the BEGIN/END lines) into the Private key data field in the GUI. Adjust the descriptive name if you want, it probably has a generic name from the upgrade process. Do not press Save yet.

We need to figure out what the serial number should be for the next certificate. This can be found by run this command from the shell:

```
# printf '%d\n' 0x`cat /root/easyrsa4pfsense/keys/serial`
```

That should return a decimal number, like ***11***, which is the serial number of the next certificate to make. Copy that number into the GUI in the Serial field, then click Save. It is important that you correct the serial number, otherwise you can end up with two certificates that have the same serial number, which will lead to problems with revocation down the road. Certificates are revoked by serial, two certs with the same serial would both be revoked if you revoke either one.

At that point you should be able to create new certificates on the Certificates tab of the Cert Manager in the GUI using this CA.

Any certificates for that CA in the GUI should also show up for use within the OpenVPN Client Export package. If you want your old/pre-existing certificates to show up there, you can import them from EasyRSA also. From the Certificates tab, click . Under Method, choose **`Import an existing certificate`**. Add a Descriptive Name (like the name of the cert). Now to get that certificate and key, go back to the shell and find the key in **`/root/easyrsa4pfsense/keys/`**. For example:

```
# cd /root/easyrsa4pfsense/keys/
# ls -l tester*
-rw-r--r-- 1 root staff 3739 Feb 3 2010 tester.crt
-rw-r--r-- 1 root staff 688 Feb 3 2010 tester.csr
```

```
-rw------- 1 root staff 887 Feb 3 2010 tester.key
```

You can then **cat** the `.crt` and `.key` files, and copy the BEGIN/.../END block that includes the encoded version into the proper place in the GUI, then click Save.

Repeat that last process for every key you want to import, and then they should all be in the GUI as well. It is not required that you have the user certificates in the GUI in order for clients to connect; They need only be there for use with the OpenVPN Client Export package.

# Chapter 9. Backup and Recovery

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 the section called “pfSense's 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. The one exception is the case of some packages, such as FreeSWITCH, which hold data outside of the configuration file.

## 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 just 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 it's a good idea to download a manual one as well. The automatic backups made on each change are good 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, it should be easy to make a habit of downloading a backup now and then, and keeping it in a safe place. If you have a subscription on portal.pfsense.org [https://portal.pfsense.org], backups can be handled easily and automatically for you with the AutoConfigBackup package.

If you make any changes to the system files, such as custom patches or code alterations, you must remember to back these changes up by hand or with the backup package described in the section called “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.

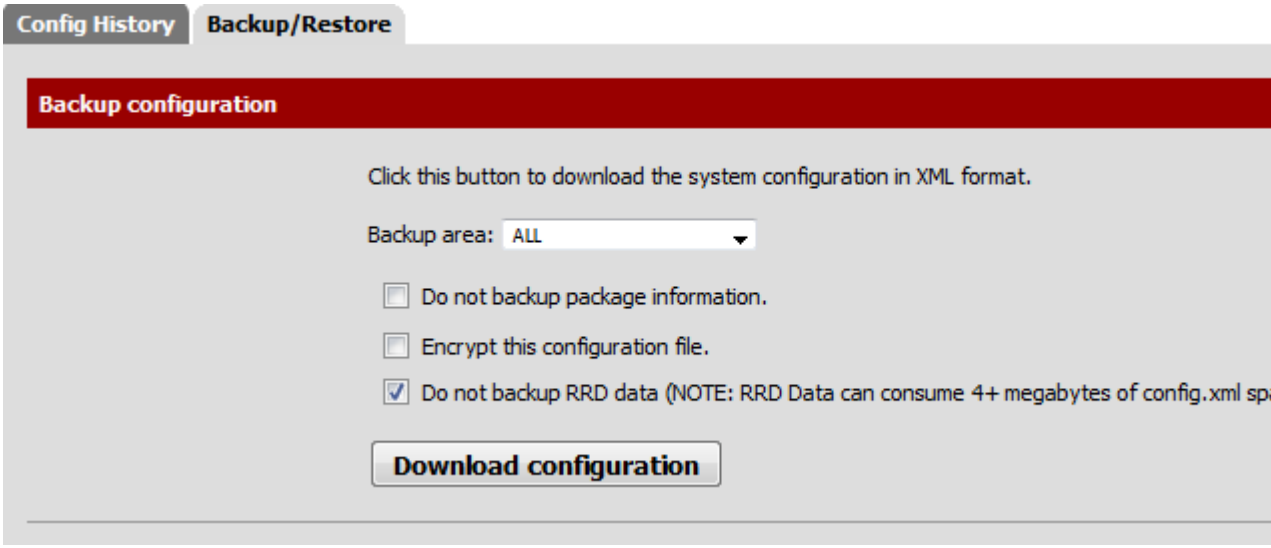
In addition to making backups, you should also test them. Before placing a system into production, you may want to backup the configuration, and then wipe the HDD, and then attempt some of the different restoration techniques in this chapter. Once you are familiar with how to both backup and restore a configuration, you may want to periodically test your backups on a non-production machine or virtual machine. The only thing worse than a missing backup is an unusable backup!

In pfSense 1.2.x, the RRD graph data, located in `/var/db/rrd`, is not backed up by default. On pfSense 2.0 and newer, the RRD data can optionally be held in the XML configuration file backup. There are also other ways to ensure this data is backed up safely. See the section called “Backup Files and Directories with the Backup Package” later in this chapter.

## Making Backups in the WebGUI

Making a backup in the WebGUI is quite simple. Just visit Diagnostics → Backup/Restore. In the Backup Configuration section of the page, ensure that Backup Area is set to **ALL**, (the default choice) then click Download Configuration (Figure 9.1, “WebGUI Backup”).

**Figure 9.1. WebGUI Backup**




Your web browser will then prompt you 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

Subscribers on portal.pfsense.org [https://portal.pfsense.org] have access to our Automatic Configuration Backup Service, AutoConfigBackup. The most up to date information on AutoConfigBackup can be found on the pfSense documentation site. [http://doc.pfsense.org/index.php/AutoConfigBackup]

## Functionality and Benefits

When you make a change to your configuration, it is automatically encrypted with the passphrase entered in your configuration, and uploaded over HTTPS to our server. Only encrypted configurations are retained on our server. This gives you instant, secure offsite backup of your firewall with no user intervention.

## pfSense Version Compatibility

The AutoConfigBackup package will work with pfSense 1.2-RELEASE and all subsequent releases including 2.x.

### Note

There is one caveat to using this package on pfSense 1.2 — the only way we could tie the automatic backup into 1.2 release is to trigger it upon every filter reload. Most page saves will trigger a filter reload, but not all.

## Installation and Configuration

To install the package, visit System → Packages and click the next to the AutoConfigBackup package. It will download and install the package. You will then find AutoConfigBackup under the Diagnostics menu.

## Setting your hostname

Make sure you have a unique hostname and domain set on the System → General Setup page. The configurations are stored by FQDN (Fully Qualified Domain Name, i.e. hostname + domain), so you must make sure each firewall you are backing up has a unique FQDN, otherwise the system cannot differentiate between multiple installations.

## Configuring AutoConfigBackup

The service is configured under Diagnostics → AutoConfigBackup. On the Settings tab, fill in your portal.pfsense.org username and password, and enter an encryption password. You should use a long, complex password to ensure your configuration is secure. For your security, we retain only encrypted configurations which are useless without your encryption password.

### Note

It is very important to store this encryption key somewhere off of your firewall — if you lose it, it will be impossible to restore your configuration if you lose the hard drive in your firewall.

## 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 the Diagnostics → AutoConfigBackup screen, and you will be shown the Restore tab, which will list your available backups along with the page that made the change (where available).

## Manually Backing Up

At times, you may want to force a backup of your configuration. You can do this on the Restore tab of the AutoConfigBackup page by clicking the Backup now button at the bottom. This will pop up a box where you can manually enter a description of your backup. You may wish to do this before making a series of significant changes, as it will leave you with a backup specifically showing the reason for the backup, which then makes it easy to revert to your configuration prior to initiating the changes. Since each configuration change triggers a backup, when you make a series of changes it can be difficult to know where you started if you should need to revert. Or you may wish to manually backup prior to upgrading to a new pfSense release, and name the backup so it's clear that is the reason you made the backup.

## Restoring Your Configuration

To restore a configuration, click the button to the right of the configuration as shown on the Diagnostics → AutoConfigBackup screen on the Restore tab. It will download the configuration specified from our server, decrypt it with your encryption password, and restore it. By default, it will not reboot. Depending on the configuration items restored, a reboot may not be necessary. For example, your firewall and NAT rules are automatically reloaded after restoring a configuration. After restoring, you are prompted if you want to reboot. If your restored configuration changes anything other than NAT and firewall rules, you should choose Yes.

# Bare Metal Restoration

If you lose your hard drive, as of now you must do the following to recover on a new installation.

1. Install pfSense on the new hard drive.
2. Bring up LAN and WAN, and assign the hostname and domain exactly the same as it was previously configured.

3. Install the AutoConfigBackup package.
4. Configure the AutoConfigBackup package as described above, using your portal account and the same encryption password as used previously.
5. Visit the Restore tab and choose the configuration you wish to restore.
6. When prompted to reboot after the restoration, do so.

You will now be back to the state of your firewall as of the last configuration change.

## Checking the AutoConfigBackup Status

You can check the success of an AutoConfigBackup run by reviewing the list of backups shown on the Restore tab. This list is pulled from our servers — if the backup is listed there, it was successfully created.

If a backup fails, an alert is logged, and you will see it scrolling across the top of the web interface.

# 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 may result in the configuration being transmitted over an untrusted link in the clear. If you must use one of these techniques, 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 should also be controlled, if not encrypted. The AutoConfigBackup package 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 could be scripted with **cron** or by some 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, you should use HTTPS with a certificate that can be verified by **wget**.

On pfSense 2.0 and later, the **wget** command must be split in two: One to login, the other to initiate the backup.

For a firewall running HTTPS with a self-signed certificate, the command would be something such as this:

```
# wget -qO/dev/null --keep-session-cookies --save-cookies cookies.txt \
       --post-data 'login=Login&usernamefld=admin&passwordfld=pfsense' \
       --no-check-certificate https://192.168.1.1/index.php
```

That will login, and then run this command to actually fetch the backup:

```
# wget --keep-session-cookies --load-cookies cookies.txt \
       --post-data 'Submit=download&donotbackuprrd=yes' https://192.168.1.1/dia
       --no-check-certificate -O config-hostname-`date +%Y%m%d%H%M%S`.xml
```

Replace the username and password with your own, and the IP address would be whichever IP address is reachable from the system performing the backup, and using HTTP or HTTPS to match your GUI. If you want to backup the RRD files, omit the `&donotbackuprrd=yes` parameter from the command.

The system performing the backup will also need access to the WebGUI, so adjust your firewall rules accordingly. Performing this over the WAN is not recommended, at a minimum you should use HTTPS, and restrict access to the WebGUI to a trusted set of public IPs. It is preferable to do this over VPN.

## Push with SCP

The configuration could also be pushed from the pfSense box 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 greatly depending on your system configuration, but may look like:

```
# 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 you would need to generate an SSH key without a passphrase. Due to the insecure nature of a key without a passphrase, generating such a key is left as an exercise for the reader. This adds some 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 highly restricted, it isn't a considerable risk in most scenarios. If you do this, 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. See the **scponly** shell 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 your 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 system's configuration. Automating this method using **cron** is also possible, but this method requires a SSH key without as passphrase on the host performing the backup. This key will enable administrative access to your firewall, so it must be tightly controlled. (See the section called “Secure Shell (SSH)” for SSH configuration details.)

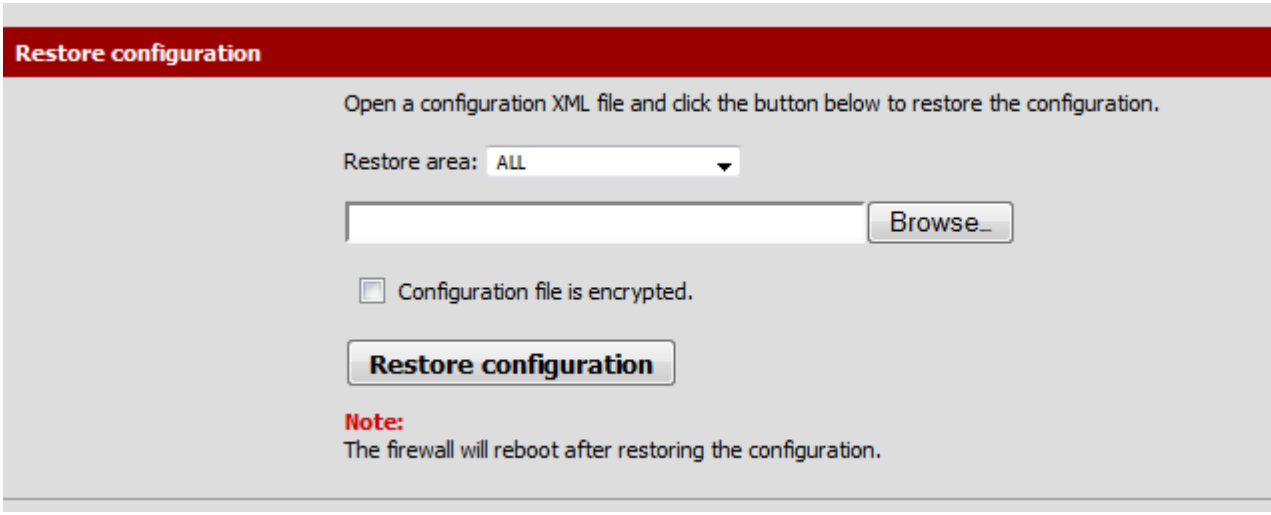
# Restoring from Backups

Backups won't do you much good 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 should have the same end result: a running system identical to what was there when the backup was made.

## Restoring with the WebGUI

The easiest way for most people to restore a configuration is by using the WebGUI. Navigate to Diagnostics → Backup/Restore, and look at the Restore configuration section (Figure 9.2, “WebGUI Restore”). To restore the backup, select the area to restore (typically **ALL**), then click Browse. Locate the backup file on your PC, and then click the Restore configuration button. The configuration will be applied, and the firewall will reboot with the settings obtained from the backup file.

**Figure 9.2. WebGUI Restore**

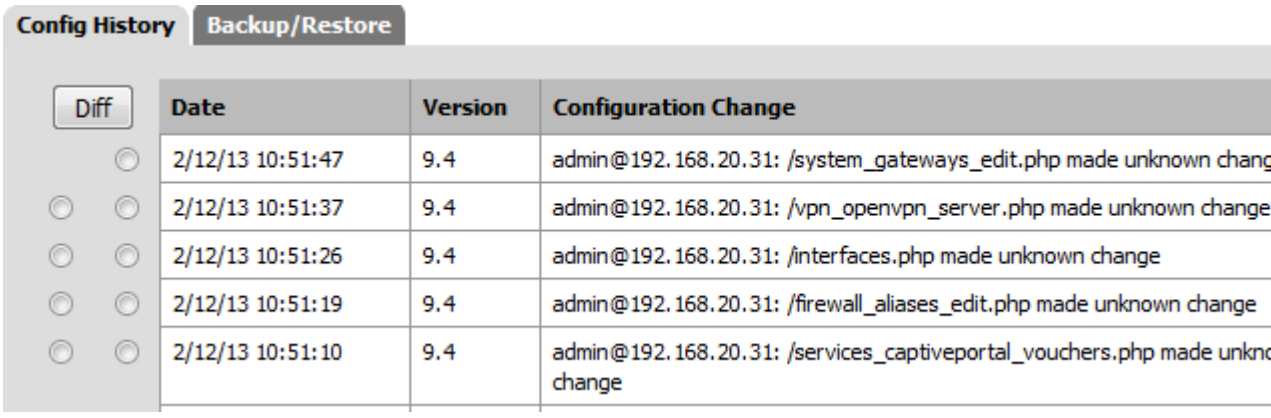



While easy to work with, this method does have some prerequisites when dealing with a full restore to a new system. First, it would need to be done after the new target system is fully installed and running. Second, it requires an additional PC connected to a working network (or crossover cable) behind the pfSense system which is being restored.

# Restoring from the Config History

For minor problems, one of pfSense's internal backups may be the easiest way to back out a change. From the Diagnostics → Backup/Restore page, click the Config History tab (Figure 9.3, “Configuration History”). The previous 30 configurations are stored, along with the current running configuration. 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.

**Figure 9.3. Configuration History**




To switch to one of these previous configurations, click the beside its entry. The configuration will be switched, but a reboot is not automatic where required. Minor changes do not require a reboot, though reverting some major changes will. To be safe, you may want to reboot the firewall with the new configuration by going to Diagnostics → Reboot System and click Yes. If a change was only made in one specific section, such as firewall rules, you may be able to simply cause 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 actions you need to take depend on what changed in the config, but the best way ensure that the full configuration is active would be a reboot.

Previously saved configurations may be deleted by clicking , but you need not delete them by hand to save space; the old configuration backups are automatically deleted when new ones are created. You may want 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 Diff

Starting with pfSense 2.0, it is now also possible to see the differences between any two configuration files 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 you have selected both files, click Diff at either the top or bottom of the column.

## Console Configuration History

Starting with pfSense 2.1, the configuration history is also available from the console menu as option **15**, Restore Recent Configuration. The command will list recent configuration files and allow you to restore them. This can be useful if a recent change has locked you out of the GUI or taken the system off the network.

# Restoring with PFI

Covered in the section called “Recovery Installation”, The Pre-Flight Installer (PFI) will take a configuration file which has been saved on a USB drive and restore it as the running configuration during the installation process. This is likely the fastest method for restoring a configuration, as it happens during the install process with no manual intervention on the pfSense box. It boots up the first time with the new configuration, and you do not need to worry about having a PC handy from which to perform the restore via the WebGUI.

# Restoring by Mounting the CF/HDD

This method is popular with embedded users. If you attach the CF or HDD of the pfSense system to a computer running FreeBSD you can mount the drive and copy a new configuration directly onto an installed system, or even copy a config from a failed system.

### Note

You can also perform this on a separate pfSense system in place of a computer running FreeBSD, but do not use an active production firewall for this purpose. Instead, use a spare system or test firewall.

The config file is kept in `/cf/conf/` for both embedded and full installs, but the difference is in the location where this directory resides. For embedded 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.

## Embedded 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`. You should also see console messages reflecting the device name, and the newly available GEOM labels.

Now mount the config partition:

```
# mount -t ufs /def/ufs/cf /mnt
```

If for some reason you are unable to use the GEOM labels, 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 should now be running with the previous configuration. If you want to copy the configuration *from* the card, the process is the same but the arguments to the **cp** command are reversed.

## Rescue Config During Install

Also covered in the section called “Recovery Installation”, this process will reinstall pfSense onto a hard drive, but maintain the configuration that is present on that drive. This is used when the contents of the system are corrupted in some way, but the configuration file is intact.

# Backup Files and Directories with the Backup Package

The Backup package will allow you to backup and restore any given set of files/folders on the system. For most, this is not necessary, but it can be useful for backing up RRD data or for packages that may have files you want to keep. To install the package, browse to System → Packages, find Backup in the list, and click . Once installed, it is available from 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 should be quite easy to make a backup of your RRD graph data (see the section called “RRD Graphs”) .

First, go to Diagnostics → Backup Files/Dir. Click  to add a new location to the backup set. In the Name field, enter **`RRD Data`**. In the Path field, enter **`/var/db/rrd`**. Set Enabled to **`True`**, and for the Description, enter **`RRD Graph Data`**. Click Save.

From the main Backup screen, click the Backup button, and then you will be presented with a file to download which should contain your RRD data along with any other directories in the backup set. Save it somewhere safe, and consider keeping multiple copies if the data is very important to you.

## Restoring RRD Data

From Diagnostics → Backup Files/Dir, click Browse, and find a backup file which was previously downloaded. Click Upload, and the files should be restored. Because the RRD files are only touched when updated once every 60 seconds, you should not have 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 your 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 should be protected as such. Hence you should protect backup copies of these files in some way. If you store them on removable media, take care with physical security of that media and/or encrypt the drive.

If you must use the WebGUI over the WAN without a VPN connection, you should at least use HTTPS. Otherwise, a backup is transmitted in the clear, including any sensitive information inside that backup file. It is highly recommended that you use a trusted link or encrypted connection.

# Chapter 10. Firewall

One of the primary functions of pfSense regardless of the role in which it is deployed is filtering traffic. This chapter covers fundamentals of firewalling, best practices, and the information you need to configure your firewall rules as necessary for your environment.

# Firewalling Fundamentals

This section deals primarily with introductory firewall concepts and lays the ground work for helping you to understand how best to appropriately configure firewall rules in pfSense.

## Basic terminology

Rule and ruleset are two terms used throughout this chapter. Rule refers to a single entry on your Firewall → Rules screen. A rule is a configuration or action for how to look at or handle network traffic. Ruleset refers to all your firewall rules as a whole. This is the sum of all the user configured and automatically added rules, which are covered further throughout this chapter.

In pfSense, rulesets on the Interface tabs are evaluated on a first match basis. This means that if you read the ruleset for an interface from top to bottom, the first rule that matches will be the one used. Evaluation stops after reaching this match and then the action specified by that rule is taken. Always keep this in mind when creating new rules, especially when you are crafting rules to restrict traffic. The most permissive rules should always be toward the bottom of the list, so that restrictions or exceptions can be made above them. The exception to this is rules on the Floating tab, but we'll get to those later.

## Stateful Filtering

pfSense is a stateful firewall. This means you only permit traffic on the interface where the traffic is initiated. When a connection is initiated matching a pass rule on your firewall, an entry is created in the firewall's state table, where information on the active connections through the firewall is retained. The reply traffic to connections initiated inside your network is automatically allowed back into your network by the state table. 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 the section called “Firewall Advanced” and the section called “State Type” 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. (See the section called “Large State Tables” about large state tables.) The default state table size in pfSense is calculated by taking about 10% of the firewall's RAM by default. On a firewall with 256MB RAM, this ends up with a state table size of 25,000. On a system with 1GB of RAM, it would be approximately 102,000.

Each user connection typically ends up with two states: One as it enters the firewall, and one as it leaves the firewall. This means if you have a state table size of 10,000, you can have around 5,000 user sessions actively traversing your firewall before any additional connections will be dropped. This limit can be increased by browsing to the System → Advanced page on the Firewall/NAT tab (Figure 10.1, “Increased state table size to 100,000”). Enter the desired number for Firewall Maximum States, or leave the box blank for the default calculated value. You can view your historical state usage under Status → RRD Graphs. On the System tab, choose **`States`** in the Graphs drop down menu.

**Figure 10.1. Increased state table size to 100,000**


# Ingress Filtering

Ingress filtering refers to the firewalling of traffic coming into your network from the Internet. In deployments with multi-WAN you have multiple ingress points. The default ingress policy on pfSense is to block all traffic, as there are no allow rules on WAN by default. Replies to traffic initiated from inside your network are automatically allowed to return through the firewall by the state table.

# Egress Filtering

Egress filtering refers to the filtering of traffic initiated inside your network destined for the Internet or any other interface on the firewall. 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 simply what most people desire. The common misperception is anything on the internal network is "trustworthy", so why bother filtering?

## Why should I 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's difficult because the administrators don't really know what is happening on the network, and are hesitant to break things. In others, it's impossible for reasons of workplace politics. But you should strive to allow only the minimum required traffic to leave your network, where possible. Tight egress filtering is important for several reasons.

1. Limit the impact of a compromised system — malware commonly uses ports and protocols that are not required on many networks. Many 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 you do not permit TCP port 6667, the usual IRC port, you can cripple bots that rely on IRC to function.

   Another example we have seen is a case where the inside interface of a pfSense install 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. It used UDP port 80, likely for a couple reasons. First, UDP allows you to send large packets 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. Second, 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. In this network, UDP port 80 was not permitted by the egress ruleset, so all the DDoS was accomplishing was pounding the inside interface of the firewall with traffic that was being dropped. I was looking at the firewall for an unrelated reason and found this; it was happily chugging along with no performance degradation and the network's administrator did not know it was happening. These types of attacks are commonly seen being 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 your firewall's state table, cost you money in bandwidth usage, and/or degrade performance for everything on your Internet connection.

Outbound SMTP is another example. You should only allow SMTP, TCP port 25, to leave your network from your mail server. Or if your mail server is externally hosted, only allow your internal systems to talk to that specific outside system on TCP port 25. This prevents every other system in your network from being used as a spam zombie, since their SMTP traffic will be dropped. This has the obvious benefit of doing your part to limit spam, and also prevents your network from being added to numerous black lists across the Internet that will prevent you from sending legitimate email to many mail servers. This may also prevent your ISP from shutting off your 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 your other measures fail.

2. Prevent a compromise — in some circumstances, egress filtering can prevent your systems from being compromised. 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. Your 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 your egress filtering and use them to his advantage.

   Again, the correct solution to preventing compromise is to fix your network's vulnerabilities, however egress filtering can help.

3. 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 messengers will port hop until finding something allowed out of your 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.

4. 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.

5. Prevent information leaks — certain protocols should never be allowed to leave your network. Specific examples will vary from one environment to another. Microsoft RPC (Remote Procedure Call) on TCP port 135, NetBIOS on TCP and UDP ports 137 through 139, and SMB/CIFS (Server Message Block/Common Internet File System) on TCP and UDP port 445 are all common examples of services that should not be allowed to leave your network. This can prevent information about your internal network from leaking onto the Internet, and will prevent your 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 in your environment are syslog, SNMP, and SNMP traps. Restricting this traffic will prevent misconfigured network devices from sending logging and other potentially sensitive information out onto the Internet. Rather than worry about what protocols might leak information out of your network and need to be blocked, solely allow the traffic that is required.

## Approaches for implementing egress filtering

On a network that has historically not employed egress filtering, it may be difficult to know what traffic is really required. This section describes some approaches for implementing egress filtering on your network.

### Allow what you know about, block the rest, and work through the fallout

One approach is to add firewall rules for the traffic you know needs to be permitted. Start with making a list of things you know are required such as in Table 10.1, “Egress traffic required”.

**Table 10.1. Egress traffic required**

| Description | Source IP | Destination IP | Destination port |
|---|---|---|---|
| HTTP and HTTPS from all hosts | any | any | TCP 80 and 443 |
| SMTP from mail server | mail server IP | any | TCP 25 |
| Recursive DNS queries from internal DNS servers | DNS server IPs | any | TCP and UDP 53 |

Then configure your firewall rules accordingly, and let everything else drop.

### Log traffic and analyze logs

Another alternative is to enable logging on your pass rules, and send the logs to a syslog server, where you can analyze them to see what traffic is leaving your network. Two log analysis packages with support for PF's logging format are fwanalog[1] and Hatchet[2]. You may find it easier to parse the logs with a custom script if you have experience with parsing text files. This will help build the required ruleset with less fallout as you should have a better idea of what traffic is necessary on your network.

# Block vs. Reject

There are two ways to disallow traffic in pfSense firewall rules — block and reject. The block setting silently drops traffic. This is the behavior of the default deny rule in pfSense, hence in a default configuration, all traffic initiated from the Internet will be silently dropped.

Reject sends a response to denied TCP and UDP traffic, letting the host that initiated the traffic know that the connection was refused. Rejected TCP traffic gets a TCP RST (reset) in response, and rejected UDP traffic gets an ICMP unreachable message in response. Though you can specify reject for any firewall rule, IP protocols other than TCP and UDP are not able to be rejected — these rules will silently drop other IP protocols. This is because there is no standard for rejecting other protocols.

## Should I use block or 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 you use 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 doesn't really hold water because every good port scanner can scan hundreds or thousands of hosts simultaneously, and isn't sitting there waiting for a response from your closed ports. There is a minimal difference in resource consumption and scanning speed, but so slight that it shouldn't be a consideration. If you block all traffic from the Internet, there is a notable difference between block and reject — nobody knows your system is actually online. If you have even a single port open, the value is minimal because the attacker knows you're online, and will also know what ports are open whether or not you reject blocked connections. While there isn't significant value in block over reject, we still recommend always using block on your WAN rules.

For rules on internal interfaces, we recommend using reject in most situations. When a host tries to access something that is not permitted in your 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, since the connection is immediately refused, it avoids these hangs. This is usually nothing more than an annoyance, but we still generally recommend using reject to avoid potential application problems

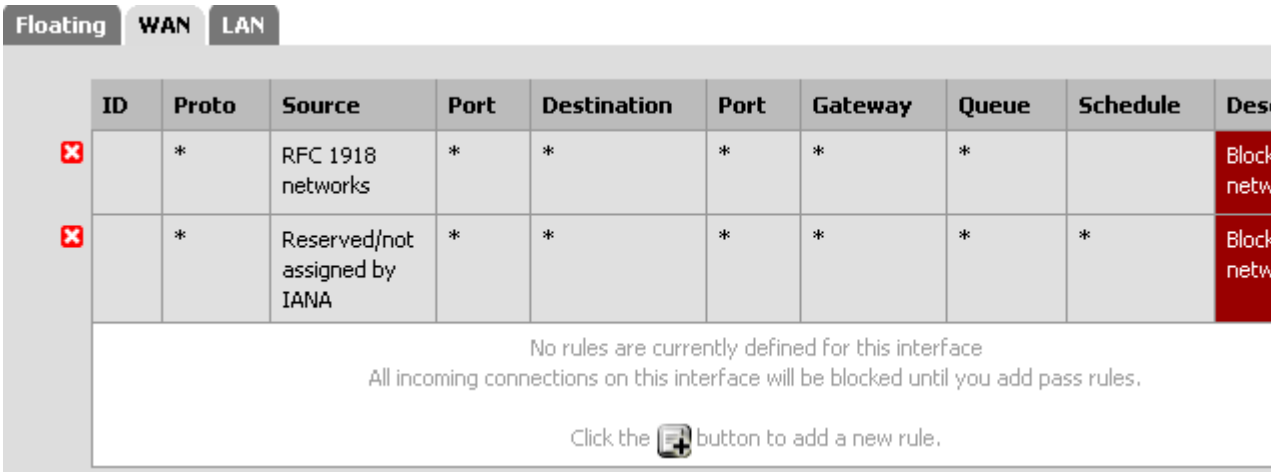
[1] http://tud.at/programm/fwanalog/
[2] http://www.dixongroup.net/hatchet/

that silently dropping traffic inside your network could induce. There is one side effect to this that may be a factor in your choice of block or reject. If you use reject, it makes it easier for people inside your network to determine your egress filtering policies as the firewall will let them know what it is blocking. It is still possible for internal users to map your egress rules when using block, it just takes a little more time and effort.

# Introduction to the Firewall Rules screen

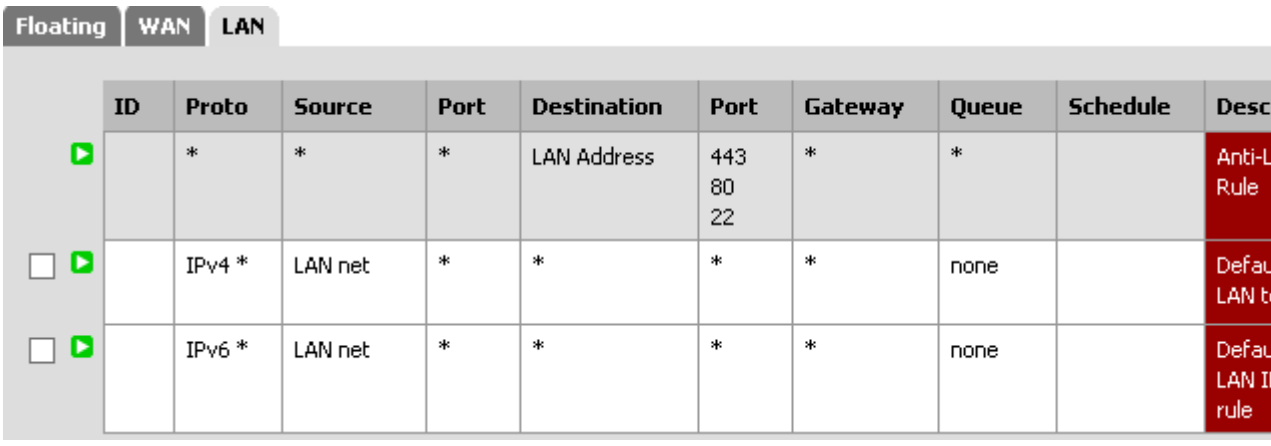
This section provides an introduction and overview of the Firewall Rules screen. First, browse to Firewall → Rules. This will bring up the WAN ruleset, which by default has no entries other than those for Block private networks and Block bogon networks if you enabled those, as shown in Figure 10.2, “Default WAN rules”. If you click the to the right of the Block private networks or Block bogon networks rules, it will take you to the WAN interface configuration page, where these options can be enabled or disabled. (See the section called “Block Private Networks” and the section called “Block Bogon Networks” for more details about blocking private and bogon networks.)

### Figure 10.2. Default WAN rules




Click on 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 10.3, “Default LAN rules”, and the Anti-Lockout Rule if you have it enabled. The anti-lockout rule is designed to prevent you from accidentally locking yourself out of the GUI. Click next to the anti-lockout rule and you will be taken to the page where you can disable the rule. For more information on how it works and how to disable it, see the section called “Anti-lockout”.

### Figure 10.3. Default LAN rules



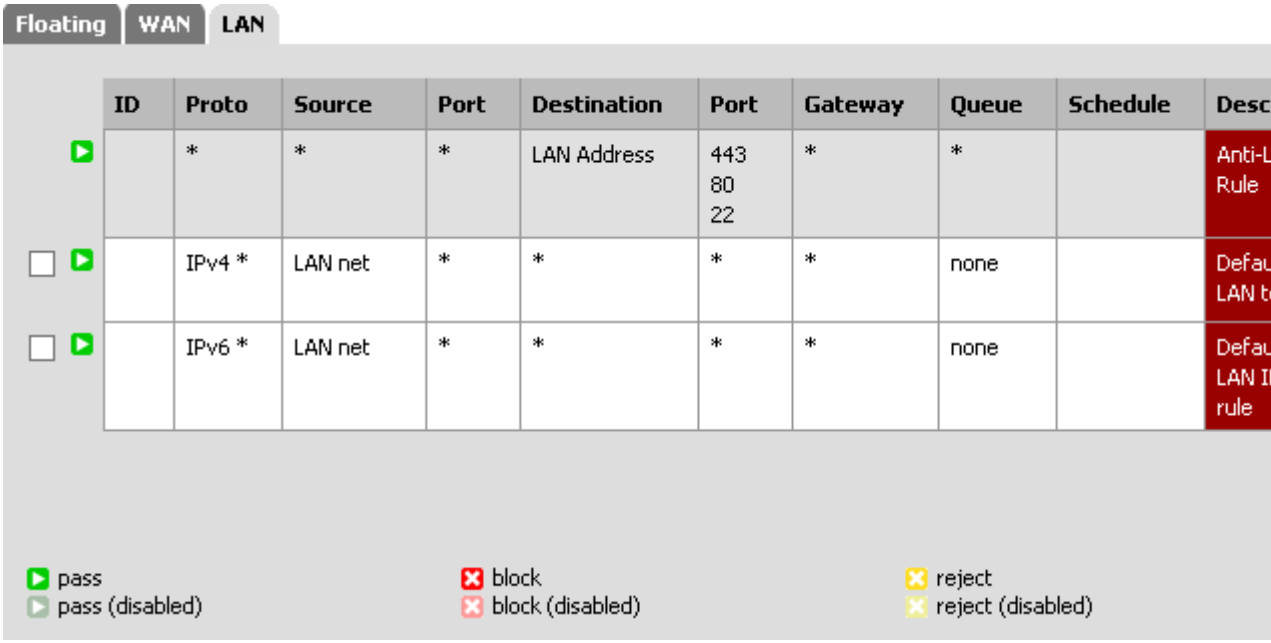
Rules for other interfaces may be viewed by clicking their respective tabs. OPT interfaces will appear with their descriptive names, so if you named your OPT1 interface DMZ, then the tab for its rules will also say DMZ.

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 there as well. The same icons are used for disabled rules, except the icon, like the rule, will be a lighter shade of their original color, or a shade of gray.

# Adding a firewall rule

Click either of the buttons on the Firewall: Rules screen to add a new rule. The top and bottom buttons, as shown in Figure 10.4, “Add LAN rule options”, will add a new rule. The top adds a rule to the top of the ruleset, while the bottom adds the rule at the bottom.

**Figure 10.4. Add LAN rule options**




If you would like to make a new rule that is similar to an existing rule, click the at the end of the row 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 rule that was just added, see the section called “Configuring firewall rules”.

# Editing Firewall Rules

To edit a firewall rule, click the to the right of the rule, or double click anywhere on the line. You will then be taken to the edit screen for that rule, where you can make any needed adjustments. See the section called “Configuring firewall rules” for more information on the options available when editing a rule.

# Moving Firewall Rules

Rules may be reordered on their own or in groups. To move rules in the list, check the box next to the rules which should be moved, or single clicking the rule will also check the box, then click the button on the row which should be underneath the relocated rules. When you hover the mouse pointer over , a thick bar will appear to indicate where the rules will be inserted. After you click , the

rules will then be inserted above the chosen row. You may also select rules to move by single clicking anywhere inside of the row you wish to select.

## Deleting Firewall Rules

To delete a single rule, click the to the right of the rule. You will be prompted to confirm the deletion, and if this is what you wanted to do, click OK to actually delete the rule.

To delete multiple rules, check the box at the start of the rows that should be removed, then click the at the bottom of the list. Rules may also be selected by single clicking anywhere on their line.

## Tracking Firewall Rule Changes

A new feature in pfSense 2.1 is that when a rule is created or updated, the user's login name, IP address, and a timestamp are added to the rule to track who added and/or last change the rule in question. If the rule existed before this feature was added, then only an update timestamp will be tracked. 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 10.5, “Firewall Rule Time Stamps”, which is visible when editing a firewall rule at the very bottom of the rule editing screen.

**Figure 10.5. Firewall Rule Time Stamps**

| Rule Information | |
|---|---|
| Created | 6/14/13 14:48:58 by **admin@192.168.20.31** |
| Updated | 7/13/13 15:41:59 by **jim@192.168.20.30** |

# Aliases

Aliases allow you to group ports, hosts, or networks and refer to them by name in your firewall rules, NAT configuration and traffic shaper configuration. This allows you to create significantly shorter, self-documenting, and more manageable rulesets. Any box in the web interface with a red background is alias friendly.

### Note

Aliases in this context should not be confused 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 screen is divided into separate tabs for each type of alias: IP, Ports, URLs, and the All tab shows every alias in one large list. When adding an alias, you can 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 IPs or hostnames |
| Network | Aliases containing CIDR-masked lists of networks, hostnames, IP ranges, or single IPs |
| 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 type alias.

URL Table — The alias is built from the file at the specified URL but is periodically updated from the URL.

Each type is described in more detail throughout this section.

# Nesting Aliases

Starting with pfSense 2.x, most aliases can be nested inside of other aliases. For example you can have an alias containing web servers, an alias containing mail servers, and then a servers alias that contains both the web and mail server aliases. URL Table aliases cannot be nested.

# Using Hostnames in Aliases

Also new in pfSense 2.x is the ability to use hostnames in aliases. You can enter any hostname into a host or network alias and it will be periodically resolved and updated behind the scenes. If a hostname returns multiple IPs, all of the returned IPs are added to the alias. This is useful for tracking DynDNS entries to allow specific users into services from dynamic IPs.

**Note**

This is not very 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 the resolve the same site name. It can work for smaller sites that have only a few servers and do not randomize their DNS responses.

# Mixing IPv4 and IPv6 Addresses in Aliases

IPv4 and IPv6 addresses may be mixed inside an alias. The appropriate type of addresses will be used when they are referenced in a specific rule.

# Alias Sizing Concerns

The total size of all tables must fix 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 your aliases, the rules may fail to load. See the section called “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.

Another similar limit is the Firewall Maximum Tables setting (the section called “Firewall Maximum Tables”), which controls the total number of aliases and other tables you can have. pfSense uses some aliases internally such as a list of NAT subnets, lockout tables, etc. The default is 3,000 and also has the same restriction as the previous setting, so that effectively limits you to around 1,500 tables.

Both of the values can be increased as much as you like, provided that you have 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 on the safe side.

# Configuring Aliases

To add an alias, go to the Firewall → Aliases screen and click the button. The following sections describe each type of alias that can be used.

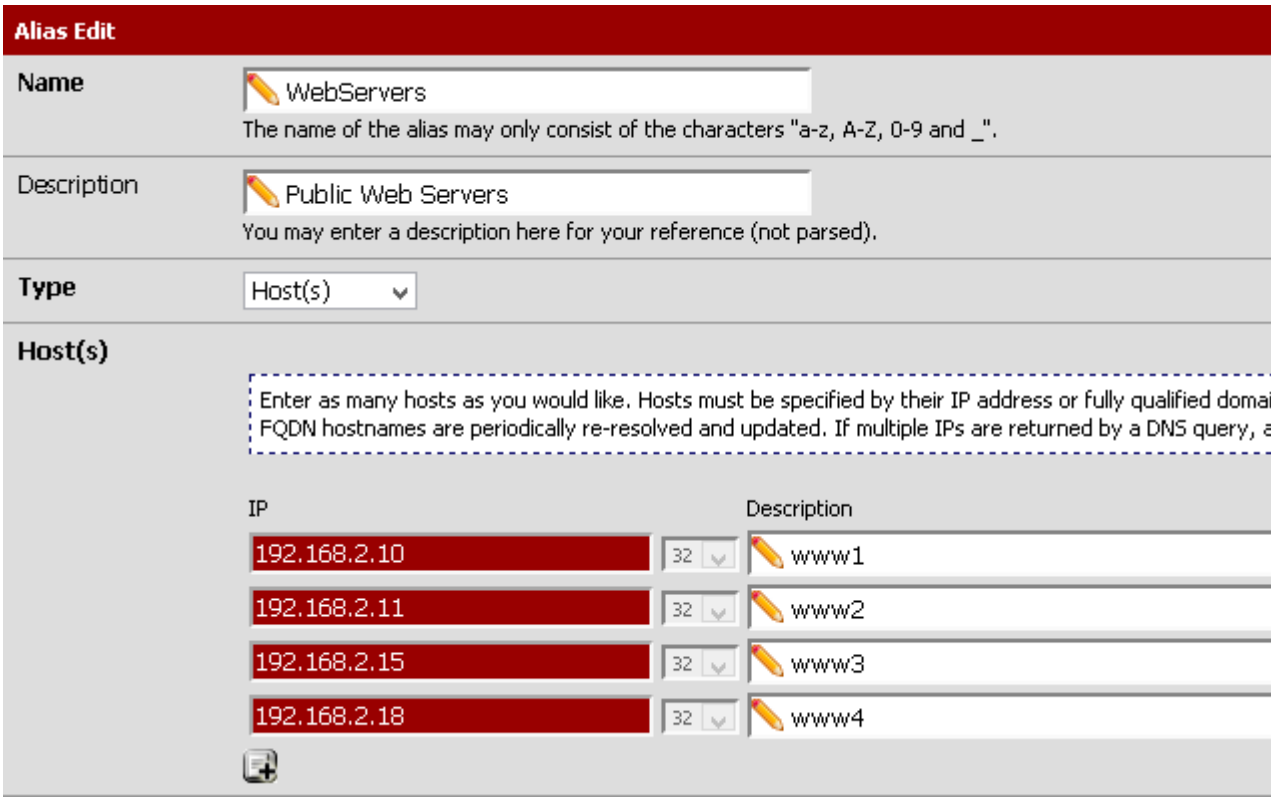
In pfSense 2.x, each manually entered alias is limited to 5,000 members, but some browsers will have trouble displaying or using the page with more than around 3,000 entries. For large numbers of entries, a URL Table type alias is recommended.

To add new members to an alias, click the at the bottom of the list of entries on the Firewall → Aliases → Edit screen.

## Host Aliases

Host aliases allow you to create groups of IP addresses. Figure 10.6, “Example hosts alias” shows an example usage of a hosts alias to contain a list of public web servers.

### Figure 10.6. Example hosts alias

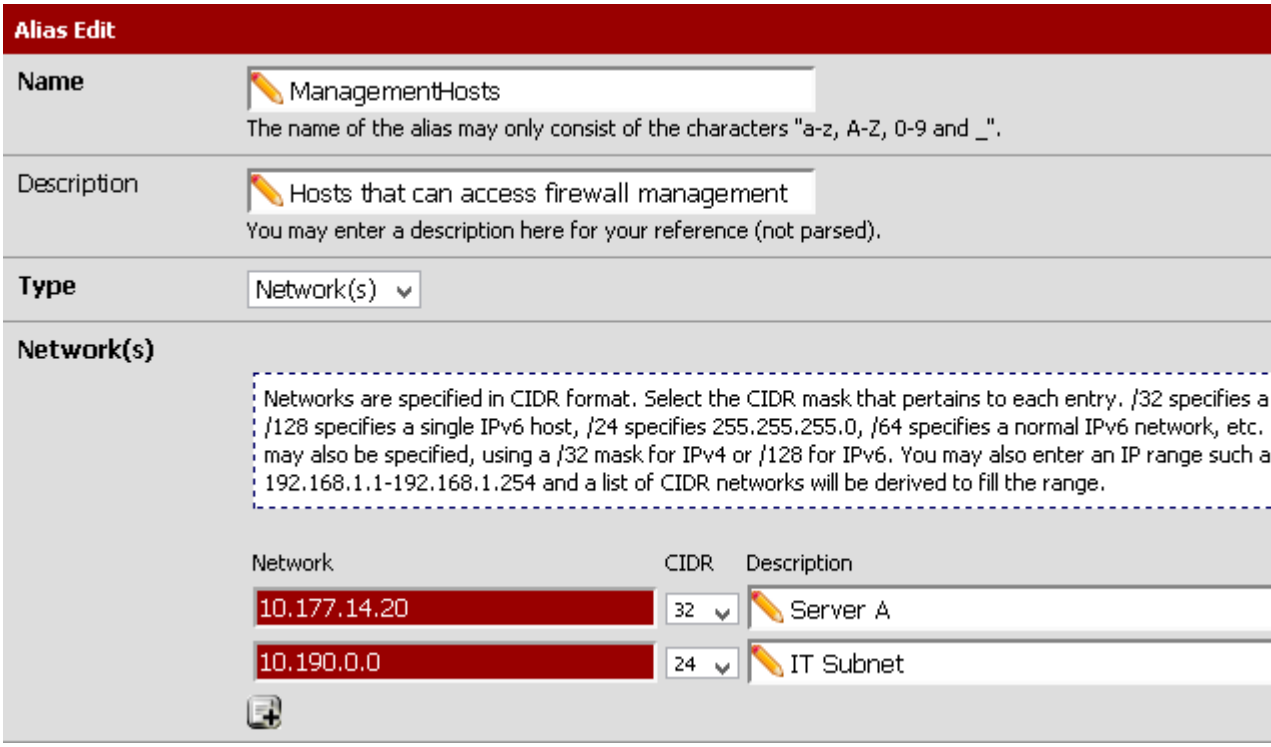



As indicated by the red background in the IP field, you may use other host aliases to nest other aliases inside this entry. You can also use hostnames as explained previously.

## Network Aliases

Network aliases allow you to create groups of networks or IP ranges. Single hosts can also be included in network aliases by selecting a /32 network mask for IPv4 addresses and /128 for IPv6 addresses. Figure 10.7, “Example network alias” shows an example of a network alias that is used later in this chapter.

### Figure 10.7. Example network alias



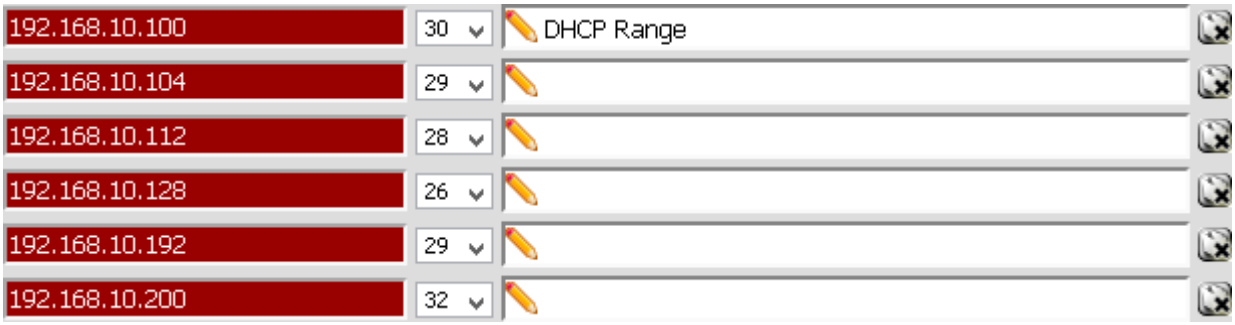

As indicated by the red background in the Network field, you may use other host or network aliases to nest other aliases inside this entry. You can also use hostnames as explained previously.

As described earlier, you can also use hostnames in the Network field and they will be resolved periodically and kept updated.

You may also enter an IPv4 range, which will be translated to an equivalent set of IPv4 CIDR networks that will exactly contain the provided range. As you can see in Figure 10.8, “Example IP Range — Before” and Figure 10.9, “Example IP Range — After”, the range is expanded when you save, and the resulting list of IPv4 CIDR networks will match exactly the range you requested, nothing more, nothing less.

### Figure 10.8. Example IP Range — Before

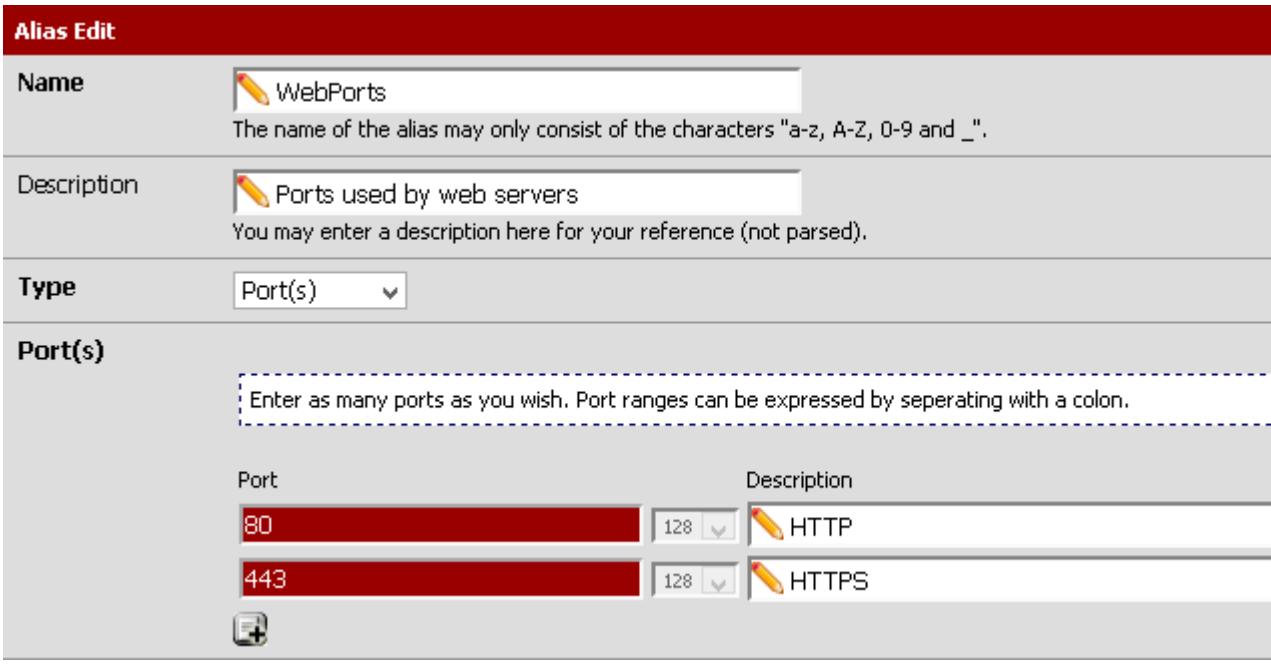

### Figure 10.9. Example IP Range — After



## Port Aliases

Port aliases enable the grouping of ports and port ranges. The protocol is not specified in the alias, rather the firewall rule where you use the alias will define the protocol as TCP, UDP, or both. Figure 10.10, “Example ports alias” shows an example of a ports alias.

**Figure 10.10. Example ports alias**




As implied by the red background, you may use aliases in the Port field to nest other port-type aliases inside this one.

## URL Aliases

With a URL alias, you specify a URL to a text file that contains a list of IP or CIDR masked network entries. Multiple URLs may be entered. When you press Save, up to 3,000 entries from each URL are read from the file and imported into a network type alias.

## URL Table Aliases

A URL Table alias behaves quite a bit differently 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 place 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 the section called “Viewing the Contents of Tables”).

For a URL Table alias, the Update Freq. drop-down controls how many days may pass before the contents of the alias are re-fetched from the stored URL. When the time comes, the contents alias will be updated overnight by a script by re-fetching the URL again.

URL Table aliases can be quite large, containing many thousands of entries. Some people 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.

# Bulk Import Network Aliases

Another method of importing multiple entries into an alias is to use the bulk import feature. To use this, on the main aliases page, click and you will be presented with a large form. First, enter an Alias Name, a Description if you like, and then in the Aliases to Import box, enter as many entries as you wish to import, separated by a carriage return. Common examples are lists of IPs, networks, and blacklists. The list may contain IP addresses, CIDR masked networks, or IP ranges.

When saved, the items are imported in full into a regular network type alias that can be edited normally.

# Using Aliases

Any box with a red background will accept an alias. When you type the first letter of an alias into any such input box, a list of matching aliases is displayed. You can select the desired alias, or type its name out completely.

## Note

Alias autocompletion is not case sensitive, as of pfSense 2.0, but it is restricted by type. For example, a Network or Host 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 10.11, “Autocompletion of hosts alias” shows how the WebServers alias configured as shown in Figure 10.6, “Example hosts alias” can be used in the Destination field when adding or editing a firewall rule. Select "Single host or alias", then type the first letter of the desired alias. Just type **W** and the alias appears as shown. Only aliases of the appropriate type are shown. For fields that require an IP address or subnet, only host and network aliases are shown. For fields that require ports, only ports aliases are shown. If there were multiple aliases beginning with "W", the drop down list that appears would show all the matching aliases.

### Figure 10.11. Autocompletion of hosts alias


Figure 10.12, “Autocompletion of ports alias” shows the autocompletion of the ports alias configured as shown in Figure 10.10, “Example ports alias”. Again if multiple aliases match the letter entered, all matching aliases of the appropriate type would be listed. You can click on the desired alias to select it.

### Figure 10.12. Autocompletion of ports alias

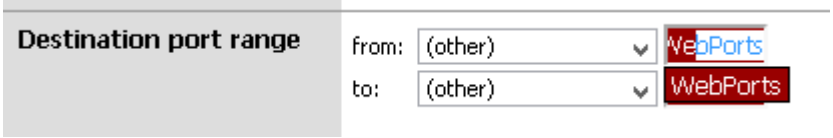
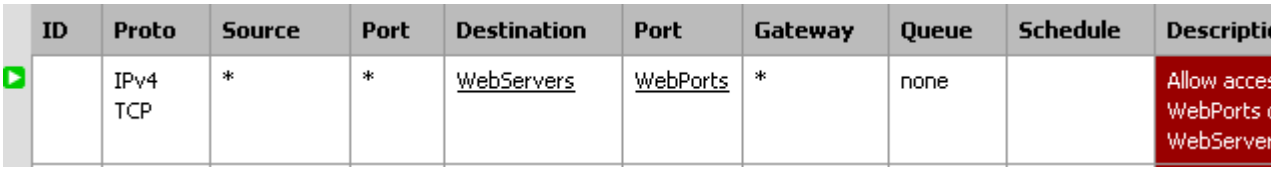



Figure 10.13, “Example Rule Using Aliases” shows the rule created using the WebServers and WebPorts aliases. This rule is on WAN, and allows any source to the IP addresses defined in the WebServers alias when using the ports defined in the WebPorts alias.

**Figure 10.13. Example Rule Using Aliases**




If you hover your mouse over an alias on the Firewall → Rules screen, a box appears showing the contents of the alias with the descriptions included in the alias. Figure 10.14, “Hovering shows Hosts contents” shows this for the WebServers alias and Figure 10.15, “Hovering shows Ports contents” for the ports alias. You can click the edit link inside the box to quickly navigate to the screen to edit the contents of the alias.

### Note

The box containing the alias information will not disappear until your mouse passes into and then out of its boundary. This behavior is necessary for the edit link to function.

**Figure 10.14. Hovering shows Hosts contents**

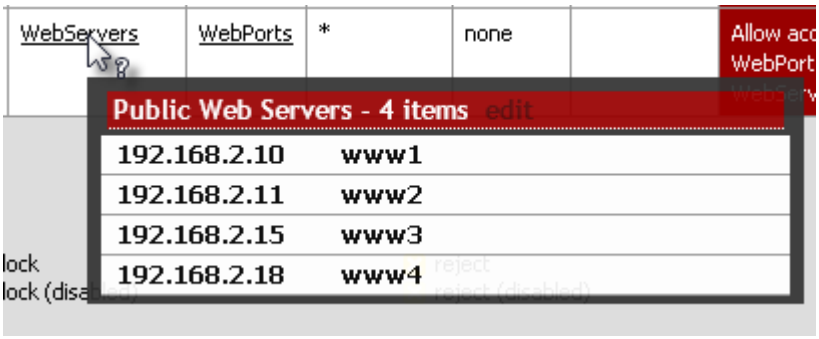
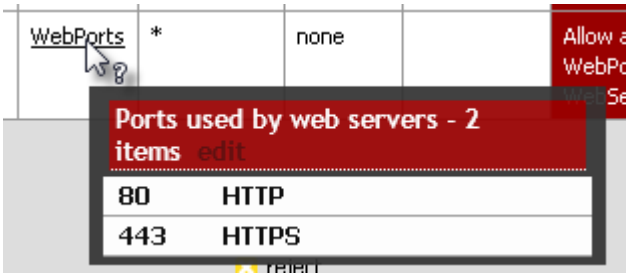



**Figure 10.15. Hovering shows Ports contents**




# Firewall Rule Best Practices

This section covers some general best practices to take into consideration when configuring your firewall.

## Default Deny

There are two basic philosophies in computer security related to access control — default allow and default deny. You should always follow a default deny strategy with your firewall rules. Configure your rules to permit only the bare minimum required traffic for the needs of your network, and let the rest drop with pfSense's built in default deny rule. In following this methodology, the number of deny rules in your ruleset should 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 uses a default deny philosophy on the WAN and a 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 want — hence is the default configuration. However it is not the recommended means of operation.

pfSense users often ask "what bad things do I need to block?" That's the wrong question, as it applies to a default permit 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.[3] Permit only what you require, 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 your 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 significantly more difficult to audit. Utilize aliases to help keep your ruleset as short as possible.

# Review your Rules

You should manually review your firewall rules and NAT configuration on a periodic basis to ensure they still match the minimum requirements of your current network environment. The recommended frequency of such review will vary 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, the configuration should be reviewed at least on a monthly basis.

Quite often when reviewing rules with customers we ask about specific rules and they say things such as "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 your Configuration

In all but the smallest networks, it can be hard to recall what is configured where and why. Use of the Description field in firewall and NAT rules is always recommended. In larger or more complex deployments, you should also maintain a more detailed configuration document describing your entire pfSense configuration. When reviewing your configuration in the future, this should help you 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 your periodic configuration reviews, it is a good idea to also review this document to ensure it remains up to date with your current configuration. You should ensure this document is updated whenever configuration changes are made.

# Reducing Log Noise

Logging is enabled on the default deny rule in pfSense by default. This means all the noise getting blocked from the Internet is going to get logged. Sometimes you won't see much noise, but in many environments you will find something incessantly spamming your logs. With connections 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,

[3] http://ranum.com/security/computer_security/editorials/dumb/index.html

among other things. You may also see your ISP's routing protocol, or router redundancy protocols such as VRRP or HSRP. In co-location environments such as data centers, you sometimes see a combination of all of those things.

Because there is no value in knowing your firewall blocked 14 million NetBIOS broadcasts in the past day, and that noise could be covering up logs that are important, it's 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 your logs.

The rule shown in Figure 10.16, “Firewall Rule to Prevent Logging Broadcasts” is one configured on one of our test systems, where the "WAN" of this test system is on an internal LAN behind an edge firewall. To get rid of the log noise so we can 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.

**Figure 10.16. Firewall Rule to Prevent Logging Broadcasts**

| ID | Proto | Source | Port | Destination | Port | Gateway | Queue | Schedule | Description |
|---|---|---|---|---|---|---|---|---|---|
| | IPv4 * | * | * | 10.0.64.255 | * | * | none | | don't log broadcasts |

You should add similar rules, matching the specifics of any log noise you are seeing in your environment. Check the firewall logs under Status → System Logs → Firewall tab to see what kind of traffic you are blocking and review its frequency. If any particular traffic is consistently being logged more than 5 times a minute, you should probably add a block rule for it to reduce your 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 should rarely be done due to the load and log levels generated. But this methodology is really a bit backwards. Blocked traffic cannot harm you 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 you are seeing significantly more or less log volume than usual, it's probably good to investigate why that is. OSSEC, an open source host-based intrusion detection system (IDS), is one system that can gather logs from pfSense via syslog and alert you to log volume abnormalities.[4]

# 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 (the section called “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, you could apply your 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 things more complex and more prone to user error, but we understand it can be desirable in specific applications.

[4] http://www.ossec.net

# Interface Groups

Interface groups, discussed in the section called “Interface Groups”, are a method to place rules on multiple interfaces at the same time. This can simplify some rule configurations if you need to place similar rules 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, PPTP, L2TP, and the PPPoE server are all actually special Interface groups that are automatically created behind the scenes.

For example, you can have a group of interfaces for all of your 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 doesn't 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 rules). The order of processing of these types is significant, and it will work like so:

1. Floating Rules
2. Interface Group Rules
3. Interface Rules

The rules are ordered in that way in the actual ruleset, so you must keep that in mind. For example, if you have an interface group rule to block traffic in a certain way, you cannot override that rule with an interface rule, because the traffic has already reached and been acted upon by the group rule.

The rules do keep processing until a match is found, however, so if something 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 you have an "allow all" rule on the OpenVPN tab, it's matched with the group rules, so the rules on the interface tab won't apply. This can be a problem if your OpenVPN rules need to have reply-to in order to ensure certain traffic exits back via the VPN.

# Automatically Added Firewall Rules

pfSense automatically adds some firewall rules, for a variety of reasons. This section describes every automatically added rule and their purpose.

## Anti-lockout Rule

To prevent locking yourself 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 your network to any protocol listening on the LAN IP.

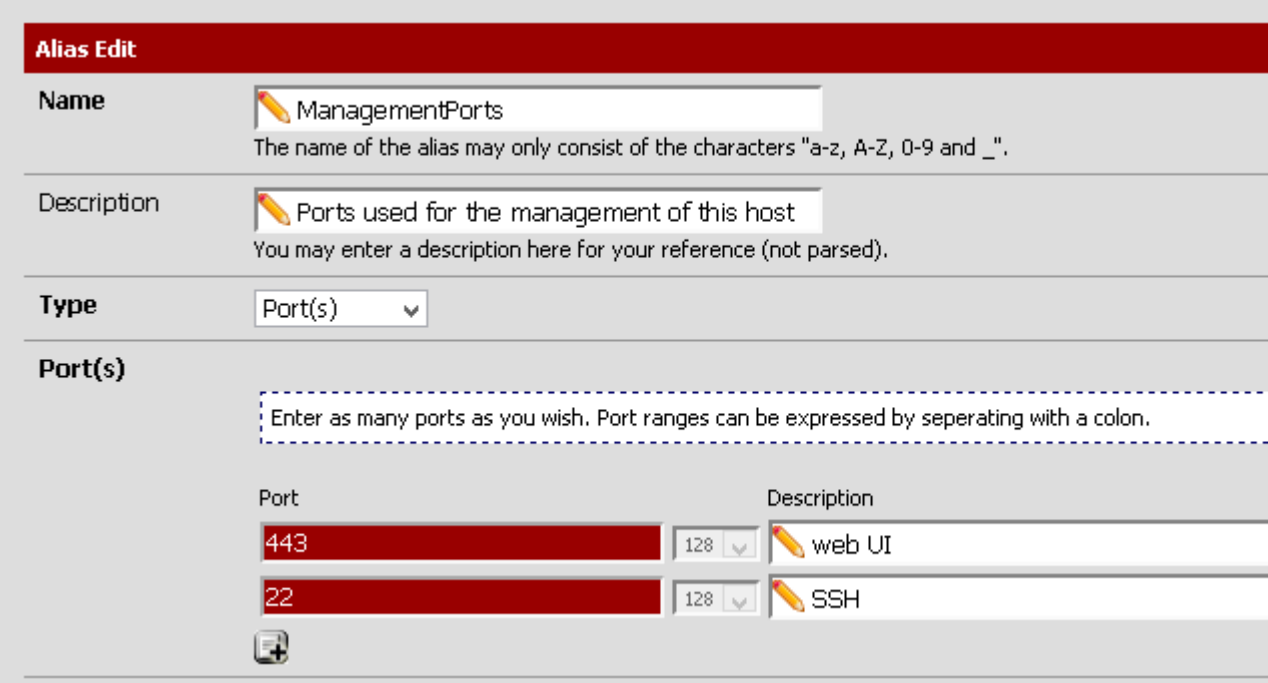
In security-conscious environments, you should disable this rule, and configure your LAN rules so only an alias of trusted hosts can access the administrative interfaces of the firewall.

### Restricting access to the administrative interface from LAN

First you need to configure the firewall rules as desired to restrict access to the management interfaces. We will walk through an example of how we usually configure this. Both SSH and HTTPS are used

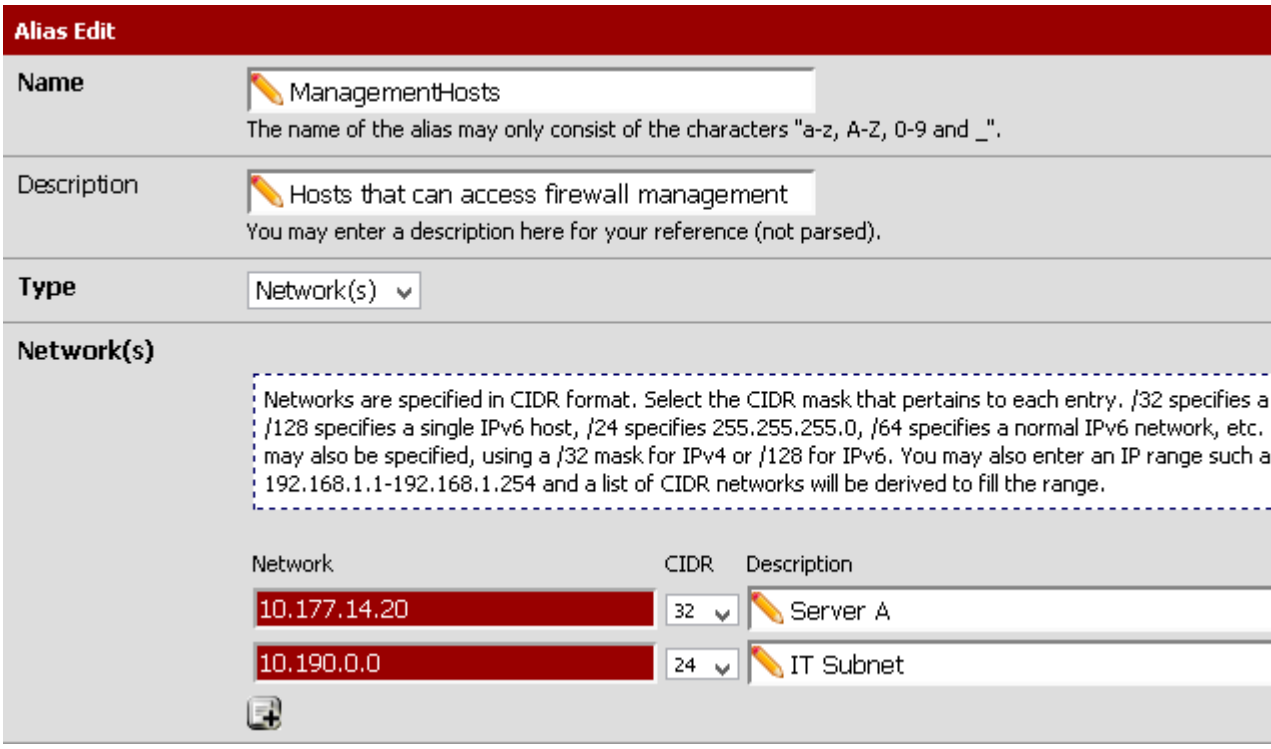
for management in this case, so create a ManagementPorts alias containing these ports (Figure 10.17, “Alias for management ports”).

### Figure 10.17. Alias for management ports




Then create an alias for hosts and/or networks that will have access to the management interfaces (Figure 10.18, “Alias for management hosts”).

### Figure 10.18. Alias for management hosts




The resulting aliases are shown in Figure 10.19, “Alias list”.

### Figure 10.19. Alias list

| Name | Values | Description |
|---|---|---|
| ManagementHosts | 10.177.14.20/32, 10.190.0.0/24 | Hosts that can access firewall management |
| ManagementPorts | 443, 22 | Ports used for the management host |

Then the LAN firewall rules must be configured to allow access to the previously defined hosts, and deny access to all else. There are numerous ways you can accomplish this, depending on specifics of your environment and how you handle egress filtering. Figure 10.20, “Example restricted management LAN rules” and Figure 10.21, “Restricted management LAN rules — alternate example” show two examples. The first allows DNS queries to the LAN IP, which is needed if you are using the DNS forwarder, and also allows LAN hosts to ping the LAN IP. It then rejects all other traffic. The second example allows access from the management hosts to the management ports, then rejects all other traffic to the management ports. Choose the methodology that works best for your environment. Remember that the source port is not the same as the destination port.

### Figure 10.20. Example restricted management LAN rules

| | ID | Proto | Source | Port | Destination | Port | Gateway | Queue | Schedul |
|---|---|---|---|---|---|---|---|---|---|
| | | IPv4 TCP/UDP | 10.0.0.0/8 | * | LAN address | 53 (DNS) | * | none | |
| | | IPv4 ICMP echoreq | 10.0.0.0/8 | * | LAN address | * | * | none | |
| | | IPv4 TCP | ManagementHosts | * | LAN address | ManagementPorts | * | none | |
| | | IPv4 * | * | * | LAN address | * | * | none | |
| | | IPv4 * | 10.0.0.0/8 | * | * | * | * | none | |

### Figure 10.21. Restricted management LAN rules — alternate example

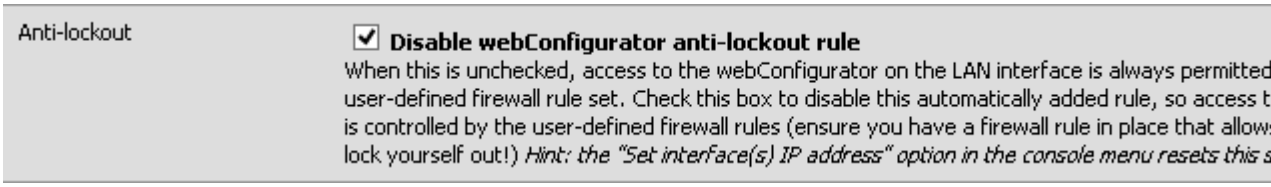
| | ID | Proto | Source | Port | Destination | Port | Gateway | Queue | Schedul |
|---|---|---|---|---|---|---|---|---|---|
| | | IPv4 TCP | ManagementHosts | * | LAN address | ManagementPorts | * | none | |
| | | IPv4 TCP | * | * | LAN address | ManagementPorts | * | none | |
| | | IPv4 * | 10.0.0.0/8 | * | * | * | * | none | |

Once the firewall rules are configured, you need to disable the webGUI anti-lockout rule on the System → Advanced page (Figure 10.22, “Anti-lockout rule disabled”). Check the box and click Save.

### Note

If you can no longer access the management interface after disabling the anti-lockout rule, you did not configure your firewall rules appropriately. You can 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. Set it to its current IP, and the rule will automatically be re-enabled.

**Figure 10.22. Anti-lockout rule disabled**




## Anti-spoofing Rules

pfSense uses PF's antispoof feature to block spoofed traffic. This provides Unicast Reverse Path Forwarding (uRPF) functionality as defined in RFC 3704 [http://www.ietf.org/rfc/rfc3704.txt]. The firewall checks each packet against its routing table, and if a connection attempt comes from a source IP on an interface where the firewall knows that network does not reside, it is dropped. For example, something coming in WAN with a source IP of an internal network is dropped. Anything initiated on the internal network with a source IP 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 you have private IP space on your WAN, you should enable this. This only applies to traffic initiated on the WAN side. You can still access hosts on private networks from the inside of your firewall. This option is now available for any interface in pfSense 2.0. You can also manually add a rule to block private networks on your OPT WAN interfaces by creating an alias containing the RFC 1918 subnets and adding a firewall rule to the top of your interface rules to block traffic with a source matching that alias. (See the section called “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. These networks should never be seen as source IPs on the Internet, and 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 you have Block bogon networks enabled, your 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 assignments are removed from the bogons list months before they are actually used, so the monthly update is adequate. If you find that you need to update the list more frequently, you may change the Update Frequency for bogons under System → Advanced on the Firewall/NAT tab.

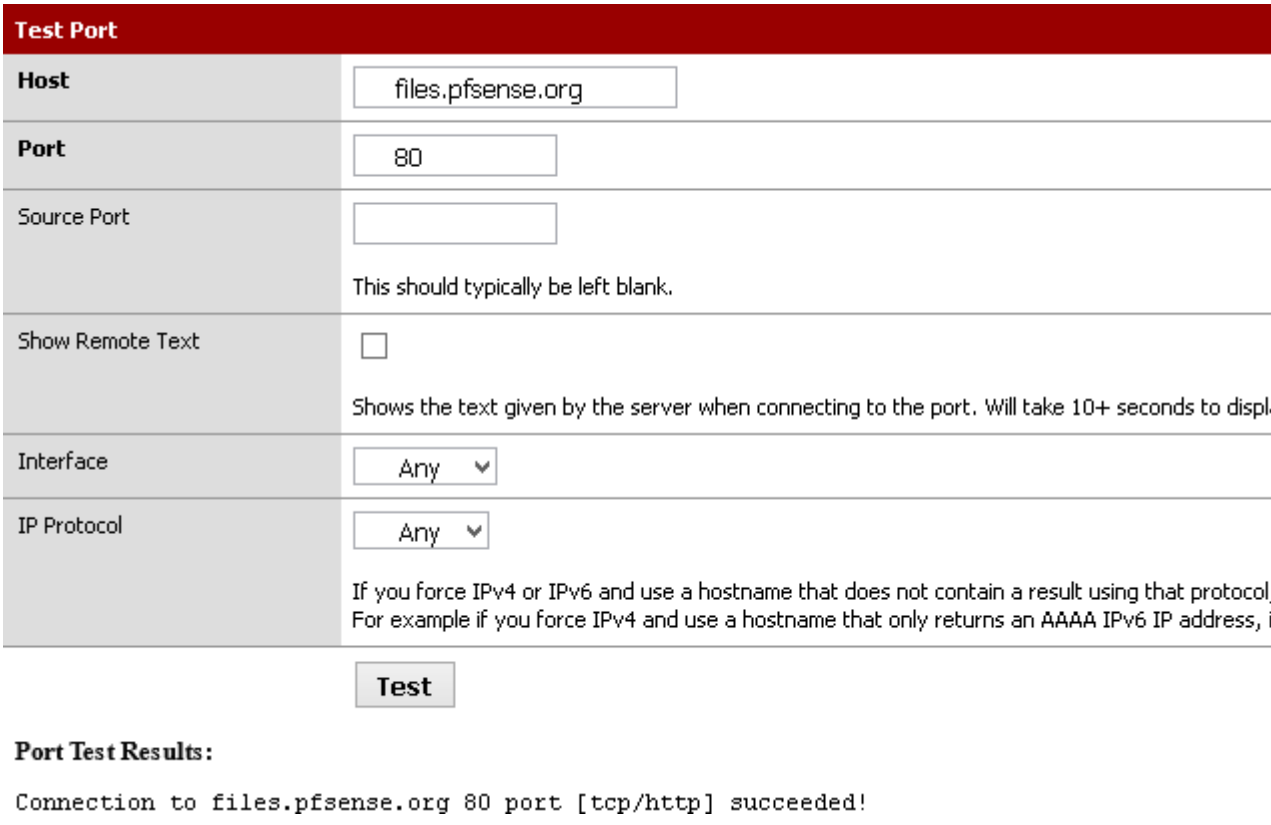
### Note

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 the section called “Firewall Maximum Table Entries” for information on changing that value.

Make sure your firewall can resolve DNS host names, otherwise the update will fail. To ensure you can resolve DNS, browse to Diagnostics → DNS, 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 10.23, “Testing connectivity for bogon updates”.

**Figure 10.23. 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, the monthly bogons update is adequate. However there may be scenarios where you want to manually force a bogon update, such as if your bogon updates have been failing because of an incorrect DNS configuration. You can execute an update via the web interface's Diagnostics → Tables screen, by selecting **bogons** or **bogonsv6** then click Download.

# IPsec

When you enable a site to site IPsec connection, 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. When you have an allow rule specifying a gateway on the inside interface containing the subnet used by the IPsec connection, and the destination of the rule is "any", a rule is automatically added to negate policy routing for traffic destined to the remote VPN subnet. If you do not want that behavior to happen, you can disable the policy route negation rules as described in the section called “Disable Negate rules” and add your own firewall rule at the top of the rules on your internal interface to pass traffic to the VPN without a gateway set.

Automatically added IPsec rules are discussed in further depth in Chapter 17, *IPsec*.

### PPTP

When you enable the PPTP server, hidden rules are automatically added allowing TCP port 1723 and the GRE (Generic Routing Encapsulation) protocol to your WAN IP address from any source IP address. More information about these rules can be found in the section called “VPNs and Firewall Rules”.

### Default Deny Rule

Rules that don't match any user-defined rules nor any of the other automatically added rules are silently blocked by the default deny rule (as discussed in the section called “Default Deny”).

# Configuring firewall rules

This section covers each individual option available on the Firewall → Rules → Edit screen when configuring firewall rules.

## Action

This is where you specify whether the rule will **`pass`**, **`block`**, or **`reject`** traffic. Each of these is covered earlier in this chapter.

## Disabled

If you wish to disable a rule without removing it from the rule list, check this box. It will still show in your firewall rules screen, but will be grayed out to indicate its disabled state.

## Interface

The Interface drop down specifies the interface on which the rule will be applied. Remember that on interface and group tab rules, traffic is only filtered on the interface where the traffic is initiated. Traffic initiated from your LAN destined to the Internet or any other interface on your firewall is filtered by the LAN ruleset.

## TCP/IP Version

Here is where you choose if this rule will apply to **`IPv4`**, **`IPv6`**, or both **`IPv4+IPv6`**. The rules will only match and act upon packets matching the correct protocol. You can use aliases that contain both types of IP addresses and the rule will match only the addresses from the correct protocol.

### Note

IPv4+IPv6 type rules can only work with ICMP, TCP, UDP, and TCP/UDP. Other protocols require separate rules for each type of traffic.

## Protocol

This is where you specify 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 make another drop down box appear where you can select the ICMP type. Several other common protocols are also available.

### Note

This field defaults to **`TCP`** for a new rule because it's a common default and it will display the expected fields for that protocol. If you wish the rule to apply to any protocol, you

will need to 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 use ping, DNS, etc.

# Source

This is where you specify the source IP address, subnet, or alias that will match this rule. You may also check the not box to negate the match.

For the Type you may specify: Any, which will match any address; Single host or alias, which will match a single IP address/hostname or alias name; or Network, which will take both an IP address and subnet mask to match a range of addresses. Lastly, there are several available presets that can be quite useful instead of entering these addresses by hand: WAN address, LAN address, LAN subnet, PPTP clients, and PPPoE users.

For rules using TCP and/or UDP, you can also specify the source port here by clicking the Advanced button. The source port is hidden behind the Advanced button because you will normally want to leave the source port 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 you should never configure it as such unless you know the application you are using employs this atypical behavior. It is also safe to define your source port as a range from 1024 to 65535.

# Destination

This is where you specify the destination IP address, subnet, or alias that will match this rule. See the description of the Source option in the section called “Source” for more details. As with the Source address setting, you may check not to negate the match.

For rules specifying TCP and/or UDP, the destination port, port range, or alias is also specified here.

# Log

This box determines whether packets that match this rule will be logged to the firewall log. Logging is discussed in more detail in the section called “Logging Practices”.

# Description

Enter a description here for your reference. This is optional, and does not affect functionality of the rule. You should enter something here describing the purpose of the rule. The maximum length is 52 characters.

# Advanced Features

Some options that are less likely to be needed or that have functionality that could be confusing to new users have been related into this section of the page. Each set of options inside this section is hidden behind an Advanced button to keep the page from being too cluttered with potentially confusing information. If an option in this section of the page has been set, then it will appear when the rule is loaded in the future without needing to click Advanced.

## Source OS

One of the more unique features of pf and hence pfSense is the ability to filter by the operating system initiating the connection. For TCP rules, pf enables passive operating system fingerprinting that allows you to create rules 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 your 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 you have technically proficient users with administrator or root level access to systems, it is possible.

## Diffserv Code Point

Differentiated Services Code Point, shortened to Diffserv Code Point or abbreviated as DSCP and sometimes referred to as the TOS field, is a way for applications to indicate inside the packets how they would prefer the 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 program or device generating the packets, for example Asterisk via its *`tos_sip`* and *`tos_audio`* configuration parameters, will set the flag in the packets and then it's up to the firewall to match and queue or act on them as needed.

If you want to match these parameters in the firewall using the Diffserv Code Point drop-down entry that matches the value set by the originating device.

The downside of DSCP is that it assumes that routers support or act on the field, which may or may not be the case, and different routers may treat the same DSCP value in unintended or mismatched ways.

### Note

This option only reads and matches the DSCP value. It does not set it.

## Advanced Options

This section lets you configure several of pf's powerful advanced options. This includes abilities to limit firewall states on a per-rule basis. By default, there are no limits or values set for any of these parameters.

### 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. IF you have IGMP or other multicast traffic with IP options that must be passed, then check this box.

### Disable Reply-To

By default, on WAN type interfaces we add *`reply-to`* in order 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 you can check this option to disable *`reply-to`* only for traffic matching this rule, rather than disabling *`reply-to`* globally.

### Marking and Matching

These are mostly useful in concert with floating rules, so you can mark a packet with a specific string on the way in with an interface rule, and then act differently on a matched packet on the way out with a floating rule. See the section called “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 IPs may simultaneously connect for this rule. Each source IP is allowed an unlimited number of connections, but the total number of source IPs allowed is restricted to this value.

### Maximum number of established connections per host

If you prefer to limit based on connections per host, this setting is what you want. Using this setting, you may limit a rule to 10 connections per source host, instead of 10 connections total. This option controls how many fully established (completed handshake) connections are allowed per host that match the rule.

### Maximum state entries per host

This setting works similarly to the established count above, but works regardless of whether or not a successful connection was made.

### Maximum new connections / per second

This method of rate limiting can help to ensure that a high connection rate will not overload a server or your state table. For example, limits can be placed on incoming connections to a mail server to reduce 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 your state table or making too many rapid connections, behaviors which are common with viruses. You can set both a connection amount and a number of seconds for the time period. Any IP address exceeding that number of connections within the given time frame will be blocked for one hour. Behind the scenes, this is handled by the virusprot table, named for its typical purpose of virus protection.

### State timeout in seconds

Here you can define a state timeout for traffic matching this rule, overriding the system's 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 the section called “Firewall Optimization Options”

#### Note

Because this only controls the traffic in the inbound direction, it is not very useful on its own. The outbound traffic will still have the default state timeout. To use this setting properly, you will also need a matching floating rule 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. If you need to account for more complex scenarios, such as working around asymmetric routing or some other non-traditional combination of traffic flow, you can alter how the flags are checked here.

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. As discussed earlier, 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 state

This is the default, and what you should almost always use.

### sloppy state

Sloppy is a less strict means of keeping state that's intended for scenarios where there is 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. Some of pf's mechanisms that prevent certain kinds of attacks will not kick in during a sloppy state check.

### synproxy state

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. You should never have a need for using this option.

**Note**

Because this only affects traffic in the inbound direction, it is not very useful, 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 CARP members. This is covered in Chapter 25, *Firewall Redundancy / High Availability*. This does not prevent a rule on a slave node from being overwritten by the master.

## 802.1p

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 you can choose how to act on it. The second will inject an 802.1p tag into a packet as it passes through this firewall. Some ISPs may require this 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

Here you can select a schedule specifying the days and times this rule will be in effect. Selecting "none" means the rule will always be enabled. For more information, see the section called “Time Based Rules” later in this chapter.

## Gateway

Gateway allows you to specify a Gateway or Gateway Group for traffic matching this rule to use. This is covered in the section called “Policy routing”.

## In/Out (Limiters)

These selections let you pick from your 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 the section called “Limiters”.

## Ackqueue/Queue

These options define which traffic shaper queues are applied to traffic entering and exiting this interface. For more information on traffic shaping, see Chapter 21, *Traffic Shaper*.

## Layer 7

Selecting an entry for Layer 7 will redirect traffic into a Layer 7 inspection instance. See the section called “Layer 7 Inspection” for more information on Layer 7 filtering and classification.

### Note

It is counter-intuitive, but rules for Layer 7 should always use the **`pass`** action. The decision to block or queue is made by the Layer 7 inspection instance, the firewall rule merely passes the traffic into the inspection daemon so it can be acted upon later.

### Note

Layer 7 cannot be used to direct traffic to a specific WAN with Multi-WAN. The classification of traffic must happen after the flow has already started, which is too late to make a routing decision.

# Floating Rules

Added in pfSense 2.0, Floating Rules are a special type of advanced rule that can perform more complicated actions than traditional interface rules. These rules can act on multiple interfaces in the inbound, outbound, or both directions. The use of inbound and outbound filtering makes things more complex and more prone to user error, but it can be desirable in specific applications.

Due to the nature of floating rules, a few additional advanced options are possible that do not exist on the normal Interface rules, and that is most of what will be covered in this section.

Most firewalls will never have any 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, users may not think to look there for firewall 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 and remote destination, similar to the source/destination in the inbound direction on a LAN-side rule.

## Potential Uses

The most common use of Floating rules is the traffic shaper rules. The rules to match and queue traffic without explicitly passing it through are possible only on the floating tab.

Another way to use floating rules is to match and control traffic leaving from the firewall itself. You can prevent the firewall from reaching specific IPs, ports, etc., or set a gateway on a floating rule to nudge traffic from the firewall out a specific WAN.

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 locations you approve, you can reduce the likelihood of later accidentally allowing traffic in through some other unintended path.

As mentioned earlier in the interface rules, you can also use them to effectively enact state timeouts, tag/match operations, and 'no state' rules.

## Processing Order

In the inbound direction, the rules work essentially the same as interface or group rules. In the outbound direction, however, things get a little more confusing.

The firewall rules are processed after NAT, so rules in the outbound direction on a WAN can never match a local/private IP source if you are using outbound NAT on that interface. By the time it hits the rule, the source address of the packet is now the WAN interface IP. It should be possible to work around this if required 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. It 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 selected.

## Quick

The quick controls whether rule processing stops when a rule is match. The quick option is added to all Interface 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".

In most situations, it is advised that you always leave quick selected. There are certain specific scenarios where leaving quick unchecked is necessary, but they are few and far between. For most, the only rules they would have without quick selected are 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 you can select one, multiple, or all possible interfaces. You can **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 just limited to the inbound direction as the interface rules are. 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, of course.

The out direction is useful for filtering traffic from the firewall itself, or matching other traffic trying to exit an interface in a way you don't want.

## Marking and Matching

You can enter a string to mark a connection on an interface tab, and then match it in the outbound direction on a floating rule here to act on it. This would be one way to act on the WAN traffic outbound from one specific internal host that you couldn't otherwise do because NAT would mask the source by the time an outbound rule on WAN saw it. It can also be used similarly for applying shaping outbound on WAN from traffic specifically tagged on the way into the firewall.

# Methods of Using Additional Public IPs

If you only have a single public IP address, you can skip to the next section. The methods of deploying additional public IP addresses will vary depending on how they are assigned, how many you have

assigned, and the goals for your network environment. To use additional public IPs with NAT, you need to configure Virtual IPs. You also have two options for directly assigning public IPs to hosts with routing public IP subnets and bridging.

# Choosing between routing, bridging, and NAT

You can either use your additional public IPs by directly assigning them on the systems that will use them, or by using NAT.

## Additional IPs via DHCP

Some ISPs force you to obtain additional IP addresses via DHCP. This is not a good means of obtaining multiple public IPs, and should be avoided in any serious network. Any business-class connection should not require doing this. We're one of the few firewalls you can use in any capacity with additional IPs from DHCP. This offers limited flexibility in what you can do with these addresses, leaving you with two feasible options.

### Bridging

If you want the additional IPs directly assigned to the systems that will use them, bridging is your only option. Use an OPT interface bridged with WAN for these systems.

### Pseudo multi-WAN

Your only option for having the firewall pull these addresses as leases is a pseudo multi-WAN deployment. Install one network interface per public IP, and configure them for DHCP. Plug all the interfaces into a switch between your firewall and your modem or router. Since you will have multiple interfaces sharing a single broadcast domain, you will want to check the box next to "This will suppress ARP messages when interfaces share the same physical network" on the System → Advanced page to eliminate ARP warnings in your logs that are normal in this type of deployment.

The only use of multiple public IPs assigned in this fashion is for port forwarding. You can configure port forwards on each WAN interface that will use the IP assigned to that interface by your ISP's DHCP server. Outbound NAT to your OPT WANs will not work because of the limitation that each WAN must have a unique gateway IP to properly direct traffic out of that WAN. This is discussed further in Chapter 15, *Multiple WAN Connections*.

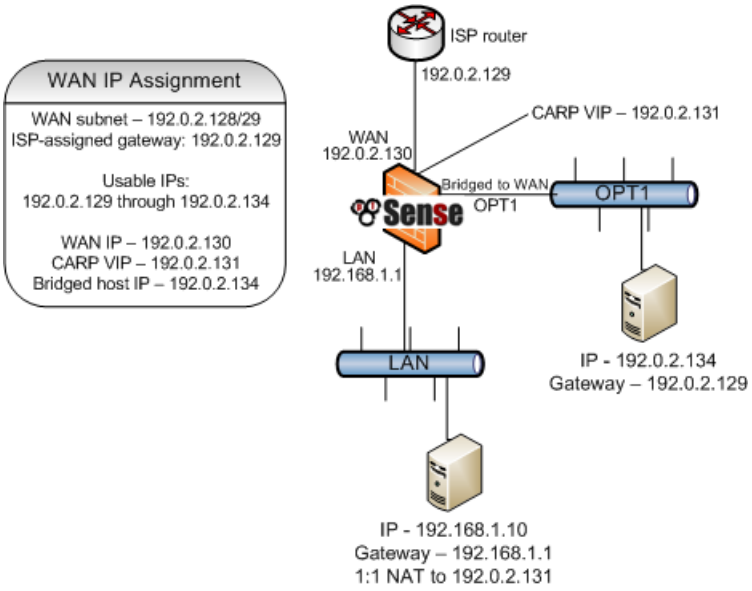
## Additional static IPs

Methods of using additional static public IPs will vary depending on the type of assignment. Each of the common scenarios is described here.

### Single IP subnet

With a single public IP subnet, one of the public IPs will be on the upstream router, commonly belonging to your ISP, with one of the IPs assigned as the WAN IP on pfSense. The remaining IPs can be used with either NAT, bridging or a combination of the two. To use them with NAT, add Proxy ARP, IP alias or CARP VIPs. To assign public IPs directly to hosts behind your firewall, you will need a dedicated interface for those hosts that is bridged to WAN. When used with bridging, the hosts with the public IPs 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 IPs need to initiate connections to hosts behind other interfaces of your firewall, since the ISP gateway will not route traffic for your internal subnets back to your firewall. Figure 10.24, “Multiple public IPs in use — single IP block” shows an example of using multiple public IPs in a single block with a combination of NAT and bridging. For information on configuration, NAT is discussed further in Chapter 11, *Network Address Translation*, and bridging in Chapter 13, *Bridging*.

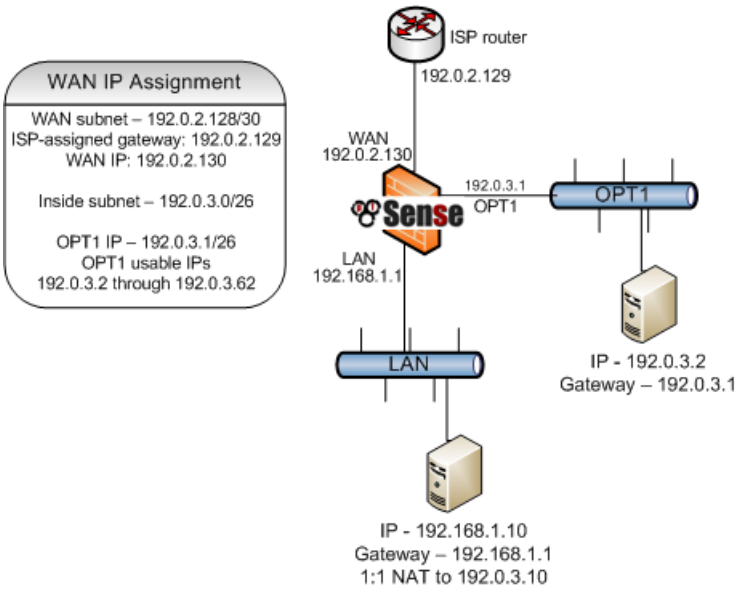
**Figure 10.24. Multiple public IPs in use — single IP block**




## Small WAN IP subnet with larger LAN IP subnet

Some ISPs will give you a small IP subnet as the "WAN side" assignment, and route a larger "inside" subnet to your end of the WAN subnet. Commonly this is a /30 on the WAN side, and a /29 or larger for use inside the firewall. The provider's router is assigned one end of the /30, typically the lowest IP, and your firewall is assigned the higher IP. The provider then routes the LAN subnet to your WAN IP. You can use those additional IPs on a routed interface with public IPs directly assigned to hosts, or with NAT using Other VIPs, or a combination of the two. Since the IPs are routed to you, ARP is not needed, and you don't need any VIP entries for use with 1:1 NAT. Because pfSense is the gateway on the OPT1 segment, routing from OPT1 hosts to LAN is much easier than in the bridged scenario required when using a single public IP block. Figure 10.25, “Multiple public IPs in use — two IP blocks” shows an example that combines a routed IP block and NAT. Routing public IPs is covered in the section called “Routing Public IPs”, and NAT in Chapter 11, *Network Address Translation*.

**Figure 10.25. Multiple public IPs in use — two IP blocks**




If you are using CARP, the WAN side subnet will need to be a /29, so each firewall has its own WAN IP, and you have a CARP IP where the provider will route the larger inside block. The inside IP subnet

must be routed to an IP 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 IP, and the provider routing to a CARP IP rather than the WAN IP. CARP is covered in Chapter 25, *Firewall Redundancy / High Availability*.

### Multiple IP subnets

In other cases, you may have multiple IP subnets from your ISP. Usually you start with one of the two previously described arrangements, and later when requesting additional IPs you are provided with an additional IP subnet. This additional subnet should be routed to you by your ISP, either to your WAN IP in the case of a single firewall, or to a CARP IP when using CARP. If your provider refuses to route the IP subnet to you, but rather routes it to their router and uses one of the IPs from the subnet as a gateway IP, you 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, your provider should route the IP subnet to you, as it makes it easier to work with regardless of your firewall of choice. It also eliminates the need to burn 3 IPs in the additional subnet, one for the network and broadcast addresses and one for the gateway IP. With a routed subnet, the entire subnet is usable where used in combination with NAT.

Where the IP subnet is routed to you, the scenario described in the section called “Small WAN IP subnet with larger LAN IP subnet” applies, just for an additional inside subnet. You can assign it to a new OPT interface, use it with NAT, or a combination of the two.

# Virtual IPs

pfSense enables the use of multiple public IP addresses in conjunction with NAT or local services through Virtual IPs (VIPs).

There are four types of Virtual IPs available in pfSense: IP Alias, CARP, Proxy ARP, and Other. Each is useful in different situations. In most circumstances, pfSense will need to provide ARP on your VIPs so you must use IP Alias, Proxy ARP or CARP. In situations where ARP is not required, such as when additional public IPs are routed by your provider to your WAN IP, use Other type VIPs.

# IP Alias

IP Alias type VIPs were added in pfSense 2.x, so they are a fairly recent addition. IP Aliases work just like any other IP address on an interface, such as the actual interface IP address. They will respond to layer 2 (ARP) and can bind services like CARP. 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 the IP Alias VIP, unless the VIP is used to forward those ports in to another device.

**Note**

IP Alias VIPs can use `Localhost` as their interface if you want to bind services using IPs from a block of routed addresses without specifically assigning the IPs to an interface. This is mostly useful in a CARP scenario so that IPs do not need to be used up by a CARP setup (one IP each per node, then the rest as CARP VIPs) when the subnet does not need to exist outside of the firewall's usage for binding services, NAT, and so on.

IP Aliases on their own do not sync to XML-RPC Configuration Sync peers because that would cause an IP 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 do synchronize. Another exception is IP Alias VIPs bound to Localhost as their interface. Because these are not active outside of the firewall node, there is no chance of a conflict so they will also synchronize.

# Proxy ARP

Proxy ARP functions strictly at layer 2, simply providing ARP replies for the specified IP address or CIDR range of IP addresses. This allows pfSense to forward traffic destined to that address according

to your 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 IPs. This is generally considered a benefit, as your additional public IPs should only be used for NAT purposes.

Proxy ARP VIPs do not sync to XML-RPC Configuration Sync peers because that would cause an IP conflict.

## CARP

CARP VIPs are mostly used with redundant deployments utilizing CARP. For information on using CARP VIPs, see Chapter 25, *Firewall Redundancy / High Availability* about hardware redundancy.

Some people prefer to use CARP VIPs even when using only a single firewall. This is usually because pfSense will respond to pings on CARP VIPs if your firewall rules permit this traffic (the default rules do not, for VIPs on WAN). Though IP Aliases may also be used for that, using CARP VIPs also prepares you for the future in case you decide to change this firewall into a redundant cluster setup.

pfSense will not respond to pings destined to Proxy ARP and Other VIPs regardless of your firewall rule configuration. With Proxy ARP and Other VIPs, you must configure NAT to an internal host for ping to function. See Chapter 11, *Network Address Translation* for more information.

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 the section called “Using IP Aliases to Reduce Heartbeat Traffic”.
- To use CARP VIPs in multiple subnets on a single interface. See the section called “Using IP Aliases to handle CARP on Multiple Subnets on a Single Interface”.

## Other

"Other" VIPs allow you to define additional IP addresses for use when ARP replies for the IP address are not required. The only function of adding an Other VIP is making that address available in the NAT configuration screens. This is useful when you have a public IP block routed to your WAN IP address, IP Alias, or a CARP VIP.

# Time Based Rules

Time based rules allow you to apply firewall rules only on specified days and/or time ranges. Time based rules are now implemented in pfSense 2.x using the pf filter, allowing time based rules to function the same as any other rule. The previous limitations of the pfSense 1.2.x schedules rules no longer apply.

## 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 you have a separate block rule 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.

It is important to always remember when using schedules that the rule will have no effect when it is no within the scheduled time. The rule will not have its action reversed because the current time is not within the scheduled time, as it was on older versions of pfSense. Keep this in mind to ensure that you do not accidentally allow more access than intended with a scheduled rule.

# Time Based Rules Caveats

There used to be limitations that prevented the use of time-based rules with captive portal or multi-wan, but those limitations no longer apply. Time-based rules can now use any of the same options as any other firewall rule.

# Configuring Schedules for Time Based Rules

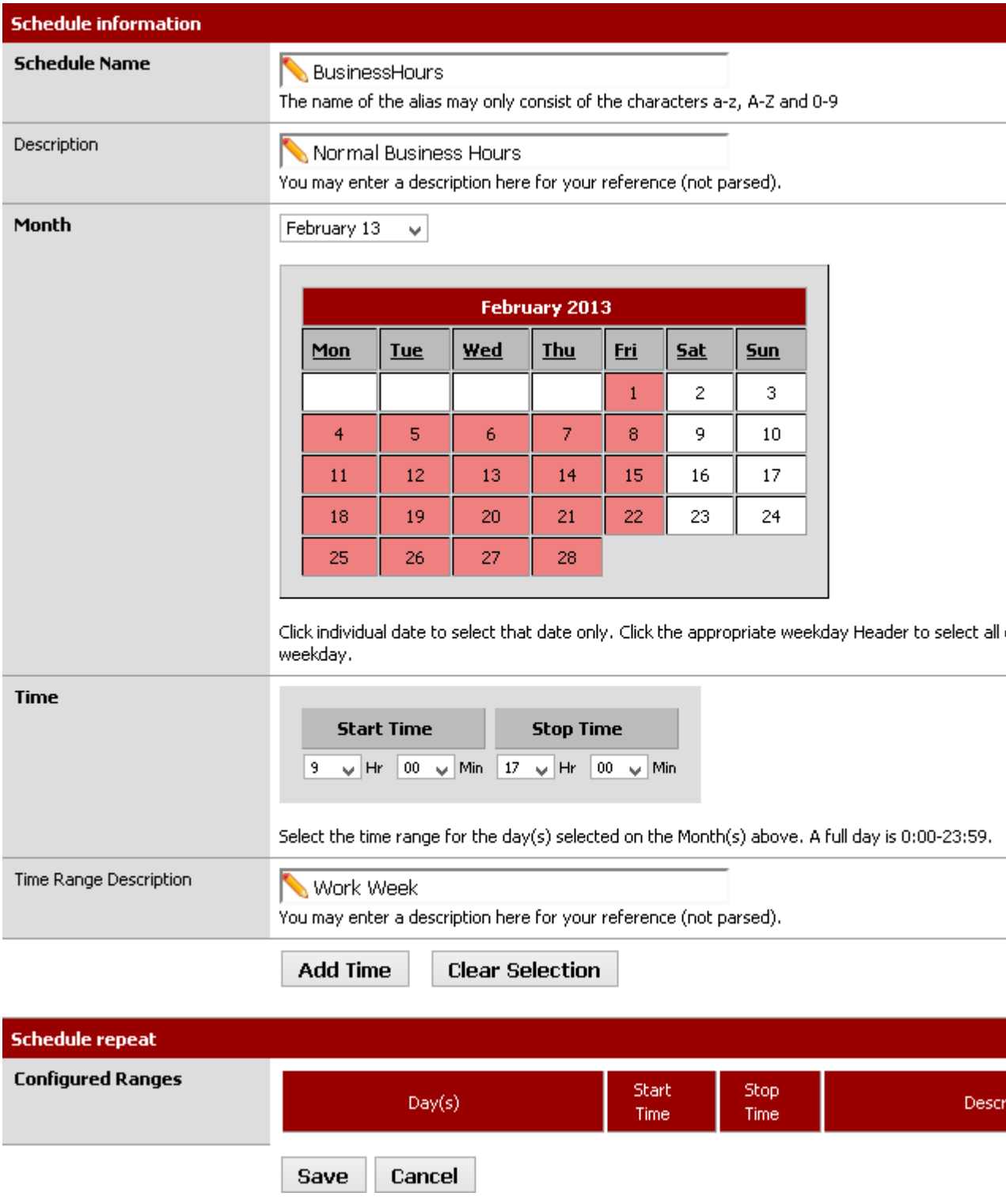
Schedules are defined under Firewall → Schedules, and each schedule can contain multiple time ranges. Once a schedule is defined, it may then be used for a firewall rule. In the following example, a company wants to deny access to HTTP during business hours, and allow it all other times of the day.

## Defining Times for a Schedule

To add a schedule from Firewall → Schedules, click . That should bring up the schedule editing screen, as seen in Figure 10.26, “Adding a Time Range”. The first field on this screen is for the Schedule Name. This setting 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, and no spaces. For this example, we'll put in **`BusinessHours`**. Next in the Description box, enter a longer free-form description of this schedule, such as **`Normal Business Hours`**. Since a schedule is made up of one or more time range definitions, you must next define a time range before you can save the schedule.

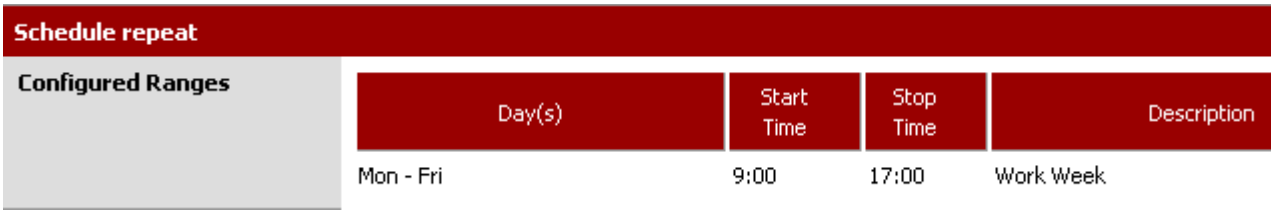
A schedule can apply to specific days, such as September 2, 2013, or to days of the week, such as Monday-Wednesday. To select any given day within the next year, choose the Month from the drop-down list, then click on the specific day or days on the calendar. To select a day of the week, click its name in the column headers. For our 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 in which this schedule should be active, in 24-hour format. Our business hours will be **`9:00`** to **`17:00`** (5pm). All times are given in the local time zone. Now enter a Time Range Description, like **`Work Week`**, then click Add Time.

**Figure 10.26. Adding a Time Range**



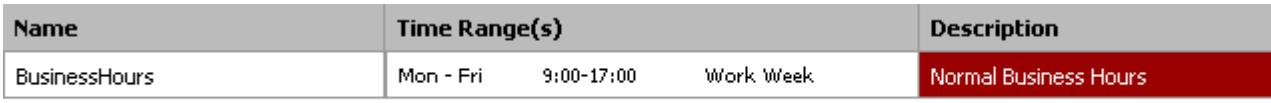

Once the time range has been defined, it will appear in the list at the bottom of the schedule editing screen, as in Figure 10.27, “Added Time Range”.

### Figure 10.27. Added Time Range




If there are more times to define, repeat that process until you are satisfied with the results. For example, 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. When all of the necessary time ranges have been defined, click Save. You will then return to the schedule list, and the new schedule will appear, as in Figure 10.28, “Schedule List after Adding”. This schedule will now be available for use in firewall rules.

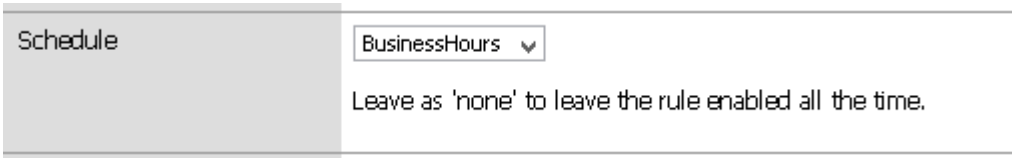
### Figure 10.28. Schedule List after Adding




## Using the Schedule in a Firewall Rule

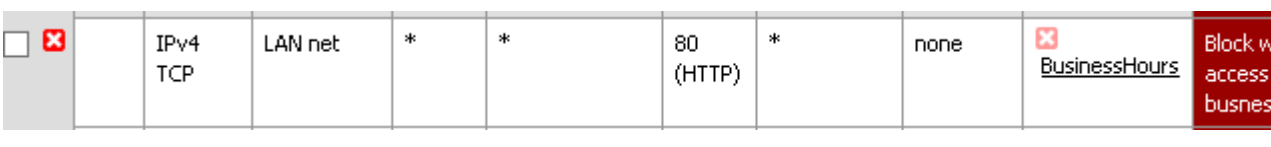
To create a firewall rule employing this schedule, you must add a rule on the desired interface. See the section called “Adding a firewall rule” and the section called “Configuring firewall rules” for more information about adding and editing rules. For our example, add a rule to block TCP traffic on the LAN interface from the LAN subnet, to any destination on the HTTP port. When you get to the Schedule setting choose the schedule we just defined, **`BusinessHours`**, as in Figure 10.29, “Choosing a Schedule for a Firewall Rule”.

### Figure 10.29. 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 you can see in Figure 10.30, “Firewall Rule List with Schedule”, this is a block rule, but the schedule column is indicating that the rule is currently not in its active blocking state because it is being viewed at a time that is outside of the scheduled range. If you hover over the schedule name, it will show the times defined for that schedule. If you hover over the schedule state indicator, it will tell you descriptively how the rule is behaving at that point in time. Since this is being viewed outside of the times defined in our BusinessHours schedule, this will say "This rule is not currently active because its period has expired". 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 during this time.

### Figure 10.30. Firewall Rule List with Schedule



Now that the rule is defined, be sure to test it both inside and outside of the scheduled times to ensure that the desired behavior is enacted.

# Viewing the Firewall Logs

For each rule that is set to log, and the default deny rule, a log entry is made. There are several ways to view these log entries, with varying levels of detail, and there is no clear "best" method.

Like other logs in pfSense, the firewall logs only keep a certain number of records. If the needs of your organization require that you maintain a permanent record of firewall logs for a longer period of time, see the section called “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 from the WebGUI, and may be found under Status → System Logs, on the Firewall tab. You can view either parsed logs, which are easier to read, or the raw logs, which have more detail if you understand PF's logging format. There is also a setting for the system logs which will show these entries in forward or reverse order. If you are unsure in which order the log entries are displayed, check the timestamp of the first and last lines, or check the section called “System Logs” for information on how to view and change these settings.

The parsed WebGUI logs, seen in Figure 10.31, “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, either pass, block, or reject. Time is the time that the packet arrived. Interface is where the packet entered pfSense. Source is the source IP address and port. Destination is the destination IP address and port. Protocol is the protocol of the packet, be it ICMP, TCP, UDP, etc.

**Figure 10.31. Example Log Entries viewed from the WebGUI**

| Act | Time | If | Source | Destination |
|---|---|---|---|---|
| | Feb 25 12:48:41 | WAN | 0.0.0.0:68 | 255.255.255.255:67 |
| | Feb 25 12:48:41 | WAN | 0.0.0.0:68 | 255.255.255.255:67 |
| | Feb 25 13:21:08 | WAN | 0.0.0.0:68 | 255.255.255.255:67 |
| | Feb 25 13:21:08 | WAN | 0.0.0.0:68 | 255.255.255.255:67 |

The action icon is a link which will lookup and display the rule which caused the log entry. More often than not, this simply says "Default Deny", but when troubleshooting rule issues it can help narrow down the suspects.

Next to the source and destination IPs you'll see a image, which is a link to perform a DNS lookup on the IP address next to the link.

If the protocol is TCP, you will also see extra fields here that represent TCP flags present in the packet. These indicate various connection states or packet attributes. Some of the more common ones are:

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

A — ACK
: Indicates ACKnowledgment of data. As discussed earlier, 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. |

There are several other flags, and their meaning is outlined in many materials on the TCP protocol. As usual, the Wikipedia article on TCP [http://en.wikipedia.org/wiki/Transmission_Control_Protocol#TCP_segment_structure] has more information.

## Easy Rule — Adding Firewall Rules from the Log View

New in pfSense 2.x is a feature called Easy Rule, which makes it easy to add firewall rules quickly from the firewall log view.

Next to the source IP address, there is an icon which, which clicked, will add a block rule for that IP address on the interface. To be more precise, it creates (or adds to) an alias containing IPs added from Easy Rule and blocks them on the selected interface.

Next to the destination IP address, there is a icon. It works similarly to the block action above, but it is more precise. It adds a pass rule that allows traffic on the interface but it must match the same protocol, source IP, destination IP, and destination port.

### Using Easy Rule to add firewall rules from the shell

It is also possible to use the shell version of Easy Rule, **easyrule**, to add a firewall from a shell prompt. If you type **easyrule**, the command will print a message explaining its usage.

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

```
# easyrule block wan 1.2.3.4
```

But to add a pass rule you need to 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 directly in real time from pf's logging interface by using option **10** from the console menu. An easy example is a log entry like that seen above in Figure 10.31, “Example Log Entries viewed from the WebGUI”:

```
Feb 25 23:59:16 pfsense pf: 00:00:00.007184 rule 3/0(match): block in on vr1: (
Feb 25 23:59:16 pfsense pf:     0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Re
Feb 25 23:59:16 pfsense pf:       Client-Ethernet-Address 3c:43:8e:86:bd:07 [|b
```

This shows that rule 39 was matched, which resulted in a block action on the `vr1` interface. The source and destination IP addresses are shown in the next line of log output. Packets from other protocols may show significantly more data.

# 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 may be quite complex and verbose. It should be relatively easy to pick out the various fields, but depending on the context of the match, it may be more difficult.

## Viewing the current contents of the log file

The filter log, as discussed in the opening on this chapter, is contained in a binary circular log so you cannot use traditional tools like **cat**, **grep**, etc. 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 current contents of the log file, run the following command:

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

The entire contents of the log file will be displayed. If you are only interested in the last few lines, you can pipe it through **tail** like so:

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

## Following the log output in real time

To "follow" the output of the clog file, you must 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.

## 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
```

You will see the log entries output one per line, with simplified output like so:

```
Feb 26 00:11:41 block vr1 UDP 0.0.0.0:68 255.255.255.255:67
```

## Finding the rule which caused a log entry

When viewing one of the raw log formats, the rule number for an entry is displayed. You can use this rule number to find the rule which caused the match. In the following example, we are trying to find out what rule is numbered *54*.

```
# pfctl -vvsr | grep '^@3 '
@3 block drop in log inet all label "Default deny rule IPv4"
```

As you can see, this was the default deny rule.

# Why do I sometimes see blocked log entries for legitimate connections?

Sometimes you will see log entries that, while labeled with the "Default deny" rule, look like they belong to legitimate traffic. The most common example is seeing a connection blocked involving a web server.

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. This happens because on occasion a packet will be lost, and the retransmits will be blocked because the firewall has already closed the connection.

It is harmless, and does not indicate an actual blocked connection. All stateful firewalls do this, though some don't generate log messages for this blocked traffic even if you log all blocked traffic.

You will see this on occasion even if you have allow all rules on all your interfaces, as allow all for TCP connections only allows TCP SYN packets. All other TCP traffic will either be part of an existing state in the state table, or will be packets with spoofed TCP flags.

A special variation of this that can indicate trouble is when you have asymmetric routing. In those cases you will mostly see TCP:SA (SYN+ACK) packets being blocked rather than FIN or RST. See the section called “Bypass Firewall Rules for Traffic on Same Interface” and the section called “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 tactics you can take, some are discussed elsewhere in the book.

## Using DNS

If using the built in DNS Forwarder, an override can be entered under Services → DNS Forwarder to resolve the website you want to block to an invalid IP (such as `127.0.0.1`).

You can also use OpenDNS for content filtering, as described in the section called “Free Content Filtering with OpenDNS”.

## Using Firewall Rules

If a website rarely changes IP addresses, access to it can be blocked using 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 IPs.

You can enter a hostname in a network alias, and then apply that alias to a block rule. On pfSense 2.x, 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.

Another option is finding all of a site's IP blocks, 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 you will need to check other regional sites as well, such as RIPE, APNIC, etc.

## Using a Proxy

If your 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 the section called “SquidGuard — Web Access Control and Filtering”.

## Prevent Bypassing Restrictions

With any of the above methods, there are many ways to get around the blocks you define. 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 content filtering capable of blocking by category, such as OpenDNS's free service which has a category for proxy sites.

To further maintain control, use a restrictive egress ruleset and only allow traffic out to services and hosts that you define. For example, only allow DNS access to the firewall or the DNS servers you explicitly want to use, such as OpenDNS. Also, if a proxy is being used, 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 on what to do if your firewall rules are not behaving as you desire or expect.

## Check your logs

Your first step when troubleshooting suspected blocked traffic should be to check your firewall logs (Status → System Logs, on the Firewall tab). Remember that by default pfSense will log all dropped traffic and will not log any passed traffic. Unless you add block or reject rules that do not use logging, all blocked traffic will always be logged. If you do not see the traffic with a red X next to it in your firewall logs, pfSense is not dropping the traffic.

## Review rule parameters

Edit the rule in question and review the parameters you have specified 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, you may need to craft a new pass rule that will match the traffic that needs to be allowed.

## Review rule ordering

Remember the first matching rule wins — no further rules are evaluated.

## Rules and interfaces

Ensure your rules are on the correct interface to function as intended. Remember traffic is filtered only by the ruleset configured on the interface where the traffic is initiated. Traffic coming from a system on your 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

It can be helpful to determine which rule is matching the traffic in question. By enabling logging on your pass rules, you can view the firewall logs and click on an individual entry to determine which rule passed the traffic.

## Troubleshooting with packet captures

Packet captures can be invaluable for troubleshooting and debugging traffic issues. You can tell if the traffic is reaching the outside interface at all, or leaving the inside interface, among many other uses. See Chapter 30, *Packet Capturing* for more details on troubleshooting with packet captures and **tcpdump**.

# Chapter 11. Network Address Translation

In its most common usage, Network Address Translation (NAT) allows you to connect multiple computers using IPv4 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 your network destined for the Internet is translated. Inbound NAT refers to traffic entering your network from the Internet. The most common type of inbound NAT and the one most are familiar with is port forwards.

In general, with the exception of Network Prefix Translation (NPt), NAT on IPv6 is not yet supported in pfSense. There is further discussion on the topic in the section called “IPv6 and NAT”. Unless otherwise mentioned, this chapter is discussing NAT with IPv4.

For those coming from pfSense 1.2.x, all of the NAT types (Port Forwards, 1:1 NAT, Outbound NAT) have gained additional functionality and have redesigned configuration screens.

# Default NAT Configuration

This section describes the default NAT configuration of pfSense. The most commonly suitable NAT configuration is generated automatically. In some environments you will want to modify this configuration, and pfSense fully enables you to do so — entirely 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

The default NAT configuration in pfSense with a two interface LAN and WAN deployment 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 the section called “Outbound NAT” about Outbound NAT.

For detecting WAN-type interfaces for use with NAT, the system looks for the presence of a gateway selected on the interface if it's static IP, or assumes it 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. If you need to allow traffic initiated on the Internet to a host on your internal network, you must configure port forwards or 1:1 NAT. This is covered in the coming sections.

# Port Forwards

Port forwards allow you to open a specific port, port range or protocol to a privately addressed device on your internal network. The name "port forward" was chosen because it is what most people

understand, and it was renamed from the more technically appropriate "Inbound NAT" after countless complaints from confused users. However it is a bit of a misnomer, as you can redirect the 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.

# Risks of Port Forwarding

In a default configuration, pfSense does not let in any traffic initiated on the Internet. This provides protection from anyone scanning the Internet looking for systems to attack. When you add a port forward, pfSense will allow any traffic matching the corresponding firewall rule. It doesn't know the difference from a packet with a malicious payload and one that is benign. If it matches the firewall rule, it's allowed. You need to rely on host based controls 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 any other services you may be running. For example this means if you allow remote web interface access from the WAN using HTTPS on TCP port 443, if you add a port forward on WAN for TCP 443 that port forward will work and your web interface access from WAN will no longer function. This does not affect access on other interfaces, just the interface containing the port forward.

# Port Forwarding and 1:1 NAT

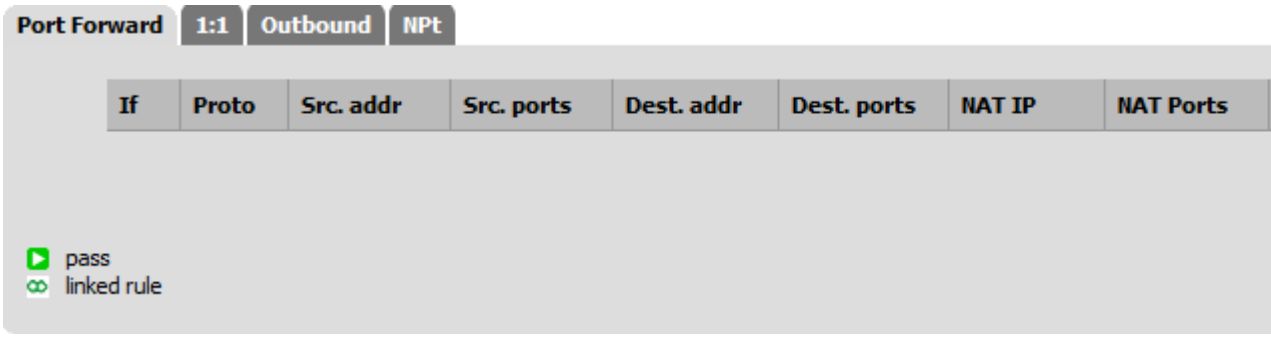
Port forwards also take precedence over 1:1 NAT. If you have a port forward on one external IP address forwarding a port to a host, and a 1:1 NAT entry on the same external IP address forwarding everything into a different host, then the port forward will still be active and working sending that one port 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 the section called “Introduction to the Firewall Rules screen”).

To begin adding a port forward entry, click the button at the very top or bottom of the list, as indicated by Figure 11.1, “Add Port Forward”.

**Figure 11.1. Add Port Forward**




You will now be looking at the Port Forward editing screen, shown in Figure 11.2, “Port Forward Example”, with the default options chosen.

The first option on the page is a checkbox to optionally Disable this NAT port forward. If you want to deactivate the rule, check this box.

The next field, 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.

The first option in most cases will be to select the Interface on which to add the port forward. In most cases this will be WAN, but if you have an OPT WAN link, or if this will be a local redirect, it may be another interface. The interface is the one where the traffic is initiated.

The Protocol 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 if you are unsure you can consult the documentation for the service or even a quick web search will likely turn up the answer. You can also use the **TCP/UDP** option to forward both TCP and UDP together in a single rule.

The Source selection is hidden behind an Advanced button by default, and set to any source. This allows you to restrict which source IPs and ports can access this port forward entry, but is not typically necessary. If it must be reachable from any location on the Internet, the source must be any. For restricted access services you can use an alias here so only a limited set of IPs may access the port forward. Unless you are absolutely certain that your service requires a specific source port, the Source Port Range should be left alone as **any**.

The Destination is set to the IP address where the traffic to be forwarded is initially destined. For port forwards on WAN, in most cases this should be set to **WAN Address**, or where you have multiple public IPs, an available Virtual IP (see the section called “Virtual IPs”) on WAN.

The Destination port range is where you specify the original destination port of the traffic, as it is coming in from the Internet, before it's redirected to the specified target host. If forwarding a single port, enter it in the from: box and leave to: blank. You can also pick from a list of common services in the drop down box available. Port aliases may also be used here to forward a set of services. If you use an alias here, the same alias must be used as the Redirect target port.

Redirect target IP is the IP where the traffic will be forwarded, or technically redirected. You may use an alias here, but the alias should 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. If you need to setup load balancing for one port to multiple internal servers, see Chapter 22, *Server Load Balancing*.

The Redirect target port is where the forwarded port range will begin. If you are forwarding a range of ports, say 19000-19100, you only specify a local starting point since the number of ports must match up one to one. This field allows you to open 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. You can also pick from a list of common services in the drop down box available. Port aliases may also be used here to forward a set of services. If you use an alias here, the same alias should be used as the Destination port range.

The description field, as in other parts of pfSense, is available for a short sentence about what the port forward does or why it exists.

If you are not using a CARP failover cluster, skip over the No XML-RPC Sync option. If you are, then checking this box will prevent this rule from being synchronized to the other members of a failover cluster (see Chapter 25, *Firewall Redundancy / High Availability*), which is usually undesirable. This option is only effective on master nodes, it does *not* prevent a rule from being overwritten on slave nodes.

NAT Reflection, covered later in this chapter in the section called “NAT Reflection”, may be enabled or disabled a per-rule basis as well to override the global default. The options in this field are explained in more detail in the section called “NAT Reflection”.

The final option is very important. The port forward entry simply defines which traffic should 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, so changes made to the NAT rule are updated in the firewall rule automatically. This is the best choice for most people. If this option is chosen, after the rule is saved a link will appear in this section of the page which takes you 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 then be manually accounted for in the firewall rule. This can be useful if you wish to set other options or restrictions 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 exists, any traffic matching this rule is forwarded in to the target system. |

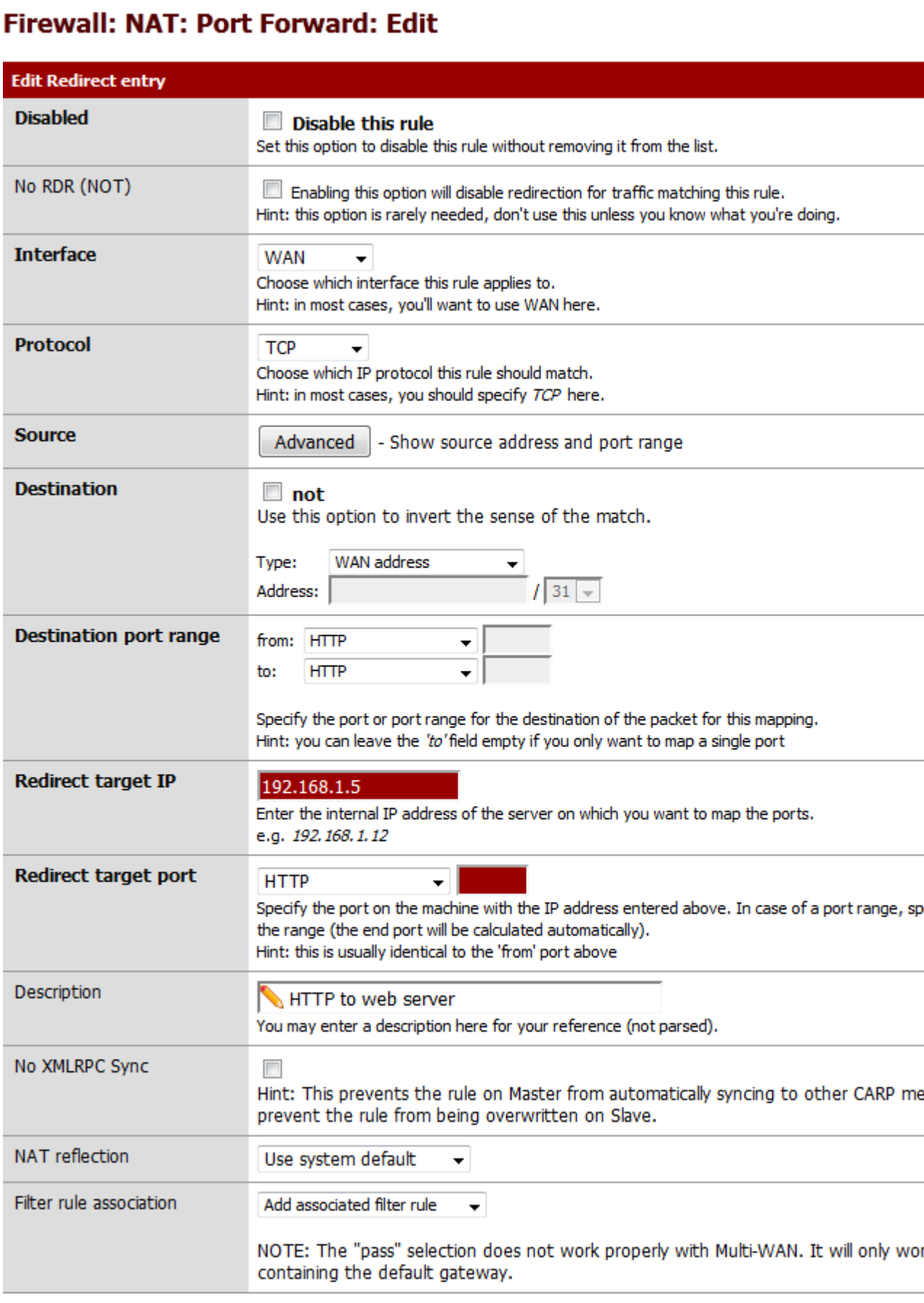
## Note

Rules using **Pass** will only work on the interface containing your default gateway, so they do not work effectively with Multi-WAN.

Click Save when finished, then Apply Changes.

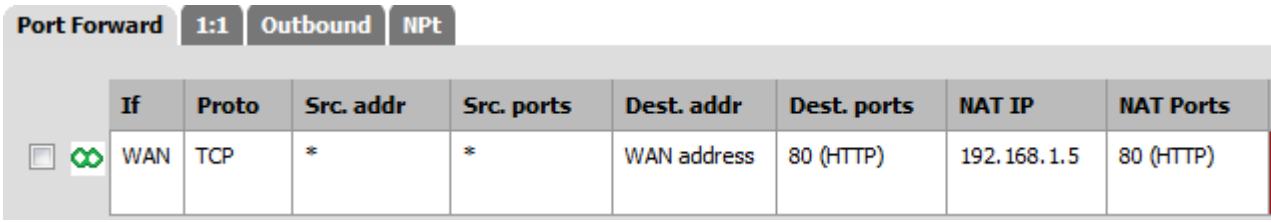
In Figure 11.2, “Port Forward Example” there is an example of the port forward editing screen filled in with the proper settings to forward HTTP inbound on WAN destined to the WAN IP to internal system 192.168.1.5.

**Figure 11.2. Port Forward Example**



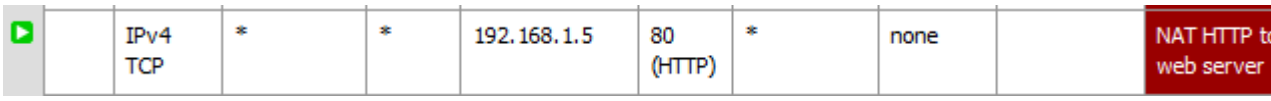
After clicking Save, you will be taken back to the port forward list, and you will see the newly created entry as in Figure 11.3, “Port Forward List”.

**Figure 11.3. Port Forward List**




You may want to double check the firewall rule, as seen under Firewall → Rules on the tab for the interface upon which the port forward was created. It will show that traffic will be allowed into the internal IP on the proper port, as shown in Figure 11.4, “Port Forward Firewall Rule”.

**Figure 11.4. Port Forward Firewall Rule**




You will want to restrict the **Source** of the automatically generated rule 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. Creating an alias of authorized hosts, and changing the source from **any** to the alias is far more secure than leaving the source wide open to the entire Internet. You may want to test first with the unrestricted source, and after verifying it works as desired, restrict the source as desired.

If everything looks right, the port forward should work when tested from outside your network. If something went wrong, see the section called “Port Forward Troubleshooting” later in this chapter.

# Tracking Changes to Port Forwards

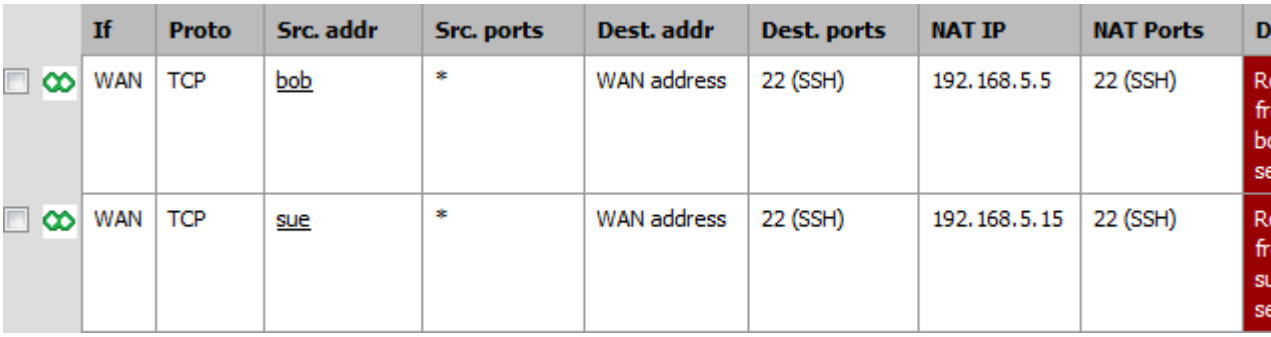
As mentioned in Figure 10.5, “Firewall Rule Time Stamps” for firewall rules, a timestamp is added to a port forward entry when it is created or last edited, to show which user created the rule, and the last person to edit the rule. Firewall rules automatically created by associated NAT rules are also marked as such on the associated firewall rule's creation timestamp.

# Port Forward Limitations

You can only forward a single port to one internal host for each public IP address you have available. For instance, you if you only have one public IP address, you can only have one internal web server that uses TCP port 80 to serve web traffic. Any additional servers would need to use alternate ports such as 8080. If you have five available public IP addresses configured as Virtual IPs, you could then have five internal web servers using port 80. See the section called “Virtual IPs” for more about Virtual IP addresses.

There is one uncommon but sometimes applicable exception to this rule. If you need to forward a particular port to a specific internal host only for certain source IPs, and forward that same port to a different host for other source IPs, that is possible by specifying the source address in the port forward entries, such as in Figure 11.5, “Port Forward Example with Different Sources”.

**Figure 11.5. Port Forward Example with Different Sources**



| | If | Proto | Src. addr | Src. ports | Dest. addr | Dest. ports | NAT IP | NAT Ports | D |
|---|---|---|---|---|---|---|---|---|---|
| | WAN | TCP | bob | * | WAN address | 22 (SSH) | 192.168.5.5 | 22 (SSH) | R fr bo se |
| | WAN | TCP | sue | * | WAN address | 22 (SSH) | 192.168.5.15 | 22 (SSH) | R fr su se |

In order for port forwards on WAN addresses to be accessible by using their respective WAN IP address from internal-facing interfaces, you will need to setup NAT reflection which is described in the section called “NAT Reflection”. You should always test your port forwards from a system on a different Internet connection, and not from inside your network, such as from a mobile device on 3G/4G.

# Service Self-Configuration With UPnP or NAT-PMP

Some programs now 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 the section called “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 your 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 SMTP to one server.

The NAT entries shown in Figure 11.6, “Example redirect port forward (negation)” and Figure 11.7, “Example redirect port forward” are an example of a configuration that will redirect all HTTP traffic coming into the LAN interface to Squid (port 3129) on the host 172.30.50.10, 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.

**Figure 11.6. Example redirect port forward (negation)**

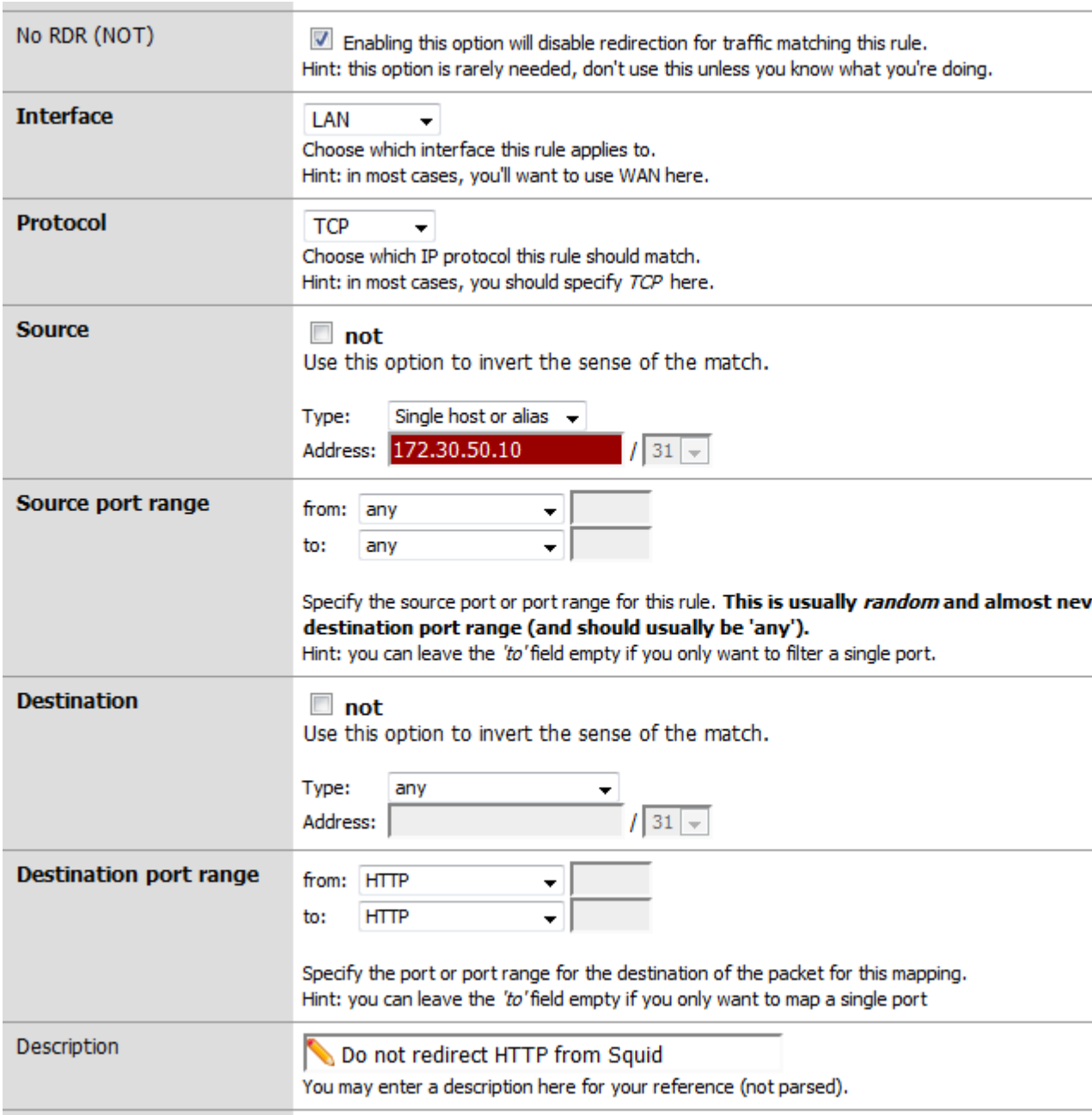
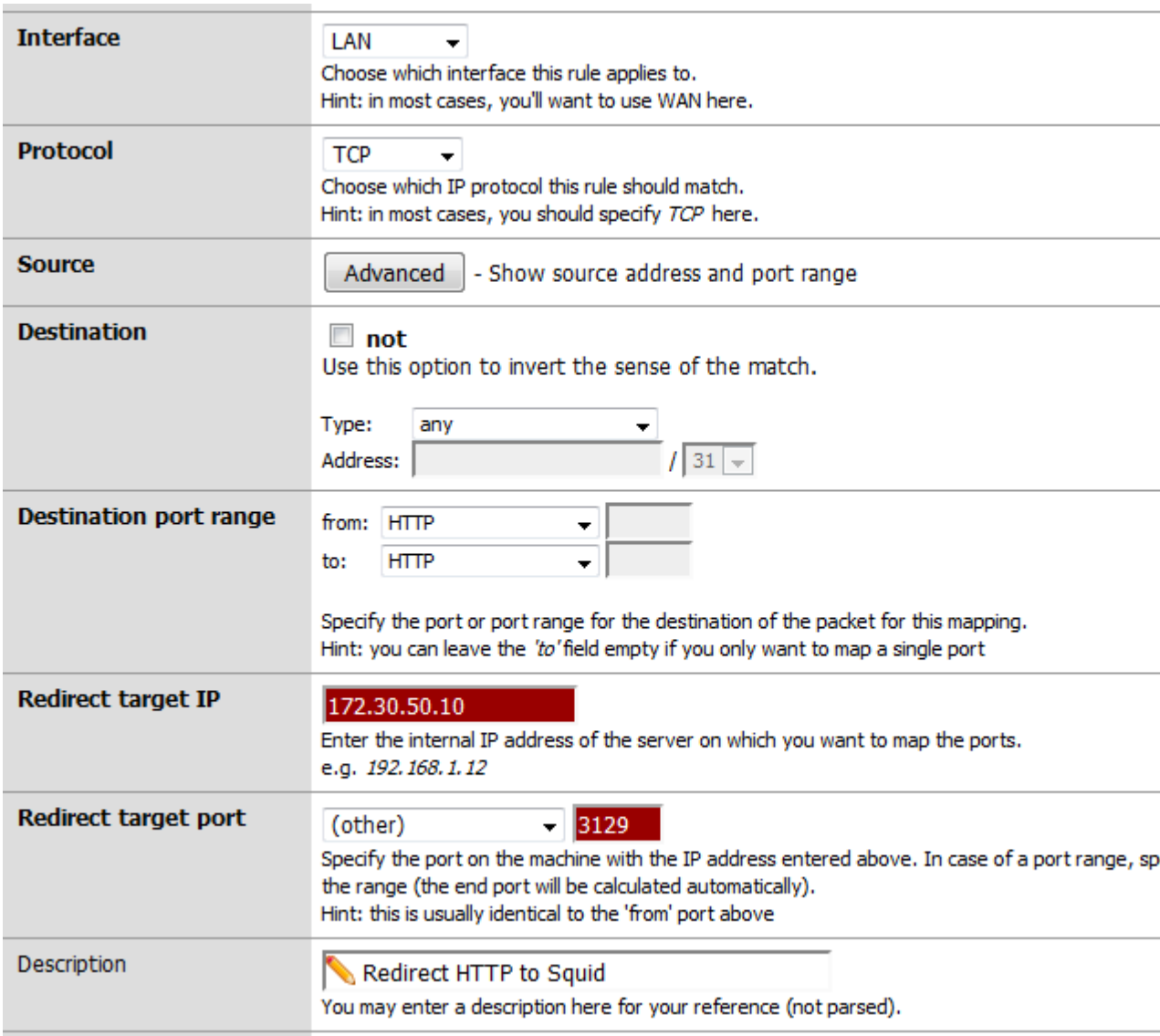


**Figure 11.7. Example redirect port forward**




# 1:1 NAT

1:1 (pronounced one to one) NAT maps one public IPv4 address to one private IPv4 address. All traffic from that private IPv4 address to the Internet will be mapped to the public IPv4 address defined in the 1:1 NAT mapping, overriding your Outbound NAT configuration. All traffic initiated on the Internet destined for the specified public IPv4 address will be translated to the private IPv4, then evaluated by your WAN firewall ruleset. If the traffic is permitted by your firewall rules to a target of the private IPv4 address, it will be passed to the internal host.

## Risks of 1:1 NAT

The risks of 1:1 NAT are largely the same as port forwards, if you allow traffic to that host in your WAN firewall rules. Any time you allow traffic, you are permitting potentially harmful traffic into your 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, you are limiting the traffic that will be allowed within the NAT rule, as well as the firewall rule. If you port forward TCP port 80, then add an allow all rule on your WAN, only TCP 80 on that internal host will be accessible. If you are using 1:1 NAT and add an allow all rule 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. Just keep this fact in mind when configuring your firewall rules, and as always, avoid permitting anything that is not required.

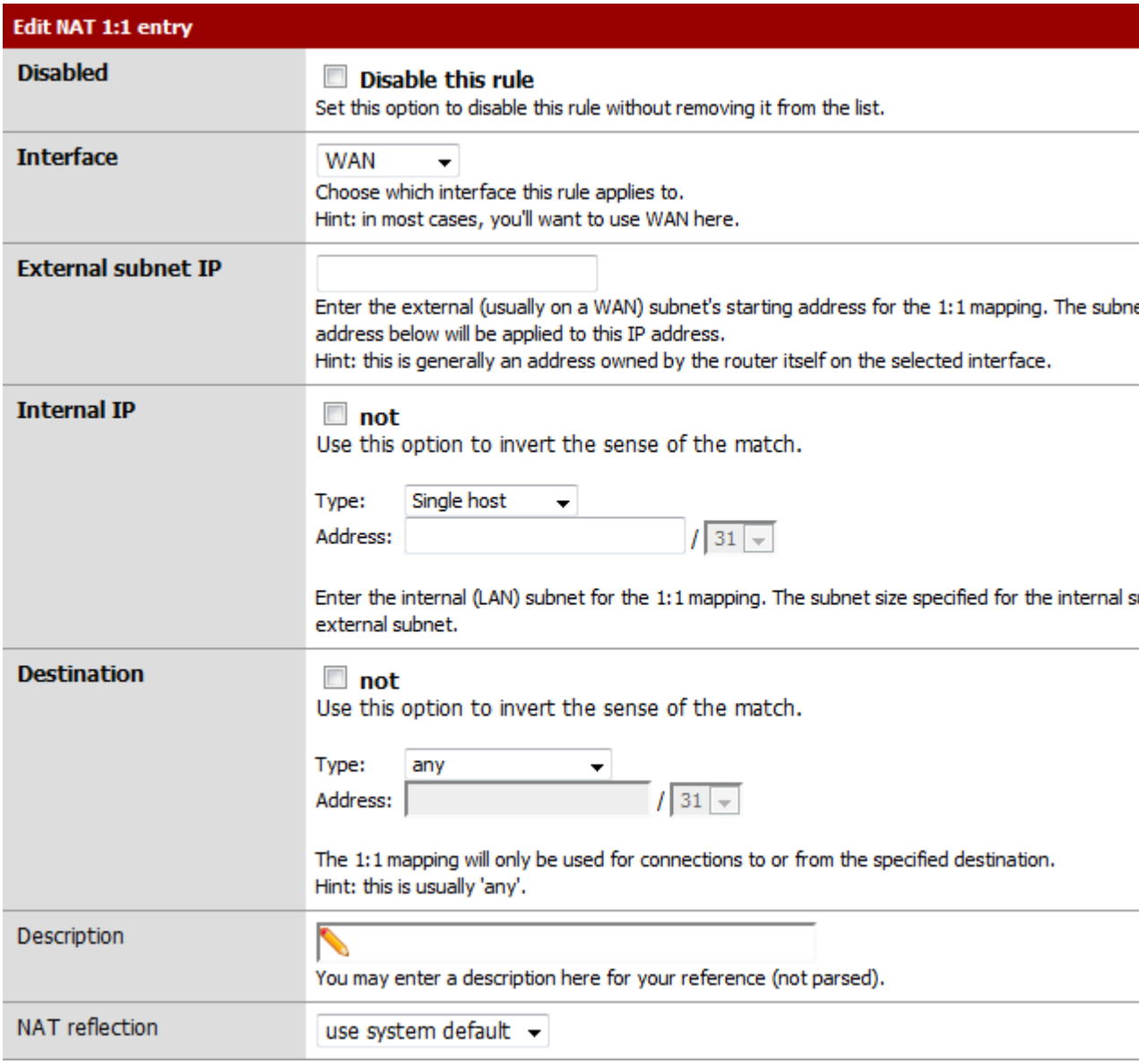
# Configuring 1:1 NAT

To configure 1:1 NAT, first add a Virtual IP for the public IP to be used for the 1:1 NAT entry as described in the section called “Virtual IPs”. Then browse to Firewall → NAT and click the 1:1 tab. Click to add a 1:1 entry.

## 1:1 NAT Entry Fields

Figure 11.8, “1:1 NAT Edit screen” shows the 1:1 NAT Edit screen, then each field will be detailed.

**Figure 11.8. 1:1 NAT Edit screen**




The 1:1 NAT screen was reorganized for pfSense 2.0, so the options may seem a little unfamiliar or confusing to some. The new options are, in order:

Disabled — Controls whether this 1:1 NAT entry is active.

| | |
|---|---|
| Interface | The interface where the 1:1 NAT translation will actually take place, typically a WAN type interface. |
| External subnet IP | The IPv4 address to which the Internal IP will be translated as it enters or leaves Interface. This is typically an IPv4 address in the Interface subnet, a VIP on Interface, or an IP address routed to the firewall on 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 that uses this firewall as its gateway, directly (attached) or indirectly (via static route). Specifying a subnet mask here will translate the entire network matching the subnet mask. For example using *x.x.x.0/24* will translate anything in that subnet to its equivalent in the external subnet. |
| Destination | Optional, a network restriction that limits the 1:1 NAT to 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. You may use an alias in the Destination field. |
| Description | An optional text description to explain the purpose of this entry. |
| NAT reflection | This option is an override for the global NAT reflection options. **Use system default** will do what the name implies, **enable** will always do NAT reflection for this entry, and **disable** will never do NAT reflection for this entry. For more information on NAT Reflection, see the section called “NAT Reflection”. |

## Example single IP 1:1 configuration

This section will show how to configure a 1:1 NAT entry with a single internal and external IP. In this example, 10.0.0.5 is a Virtual IP on the WAN. In most deployments this will be substituted with one of your public IP addresses. The mail server being configured for this mapping resides on a DMZ segment using internal IP 192.168.2.5. The 1:1 NAT entry to map 10.0.0.5 to 192.168.2.5 is shown in Figure 11.9, “1:1 NAT Entry”. A diagram depicting this configuration is in Figure 11.10, “1:1 NAT Example — Single inside and outside IP”.

**Figure 11.9. 1:1 NAT Entry**

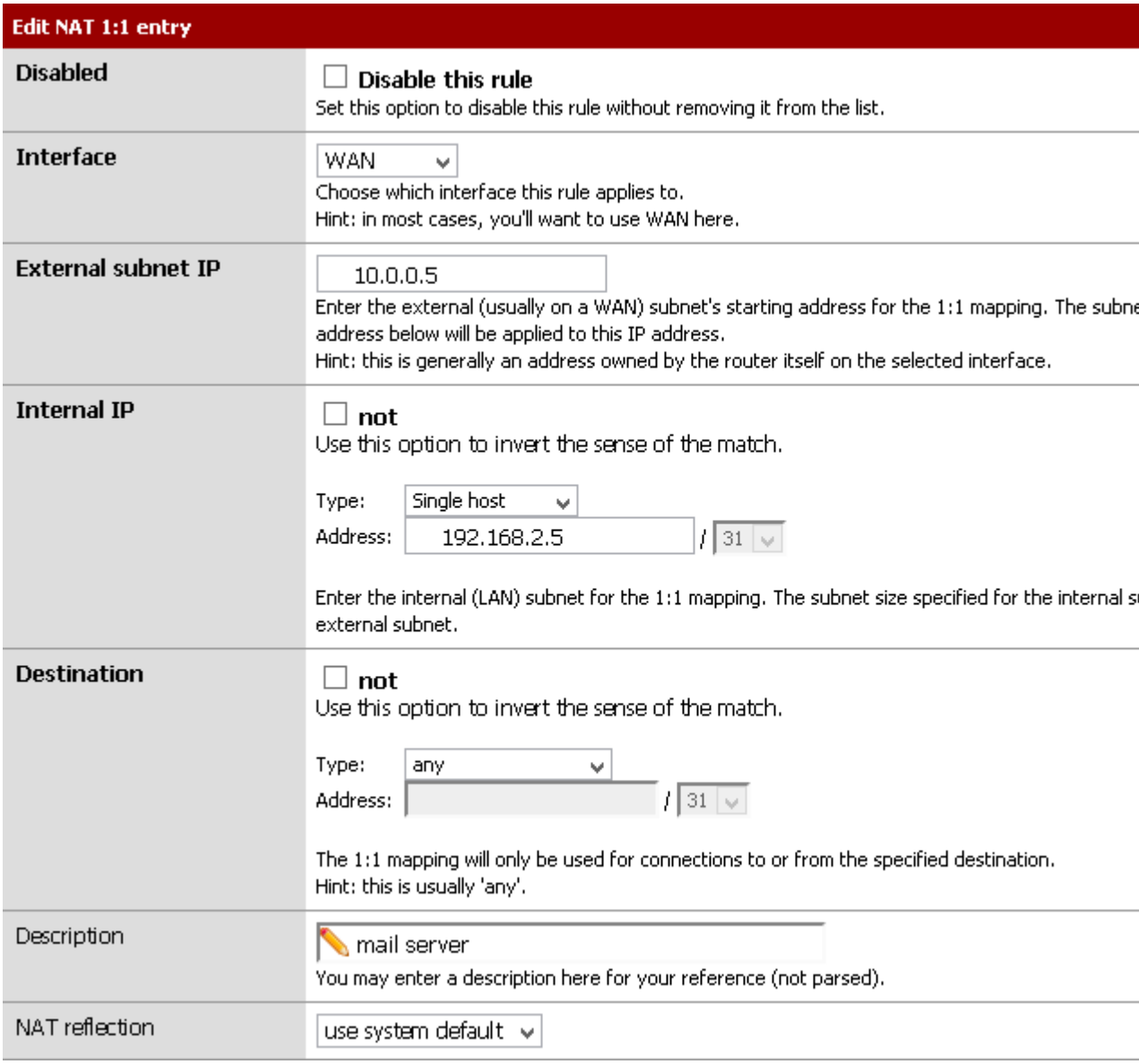
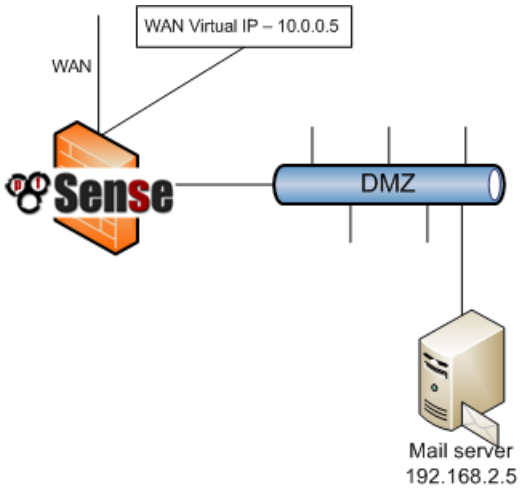



**Figure 11.10. 1:1 NAT Example — Single inside and outside IP**



## Example IP range 1:1 configuration

1:1 NAT can be configured for multiple public IPs by using CIDR ranges. CIDR summarization is covered in the section called “CIDR Summarization”. This section covers configuration of 1:1 NAT for a /30 CIDR range of IPs.

### Table 11.1. /30 CIDR mapping — matching final octet

| External IPs | Internal IPs |
|---|---|
| **10.0.0.64/30** | **192.168.2.64/30** |
| 10.0.0.64 | 192.168.2.64 |
| 10.0.0.65 | 192.168.2.65 |
| 10.0.0.66 | 192.168.2.66 |
| 10.0.0.67 | 192.168.2.67 |

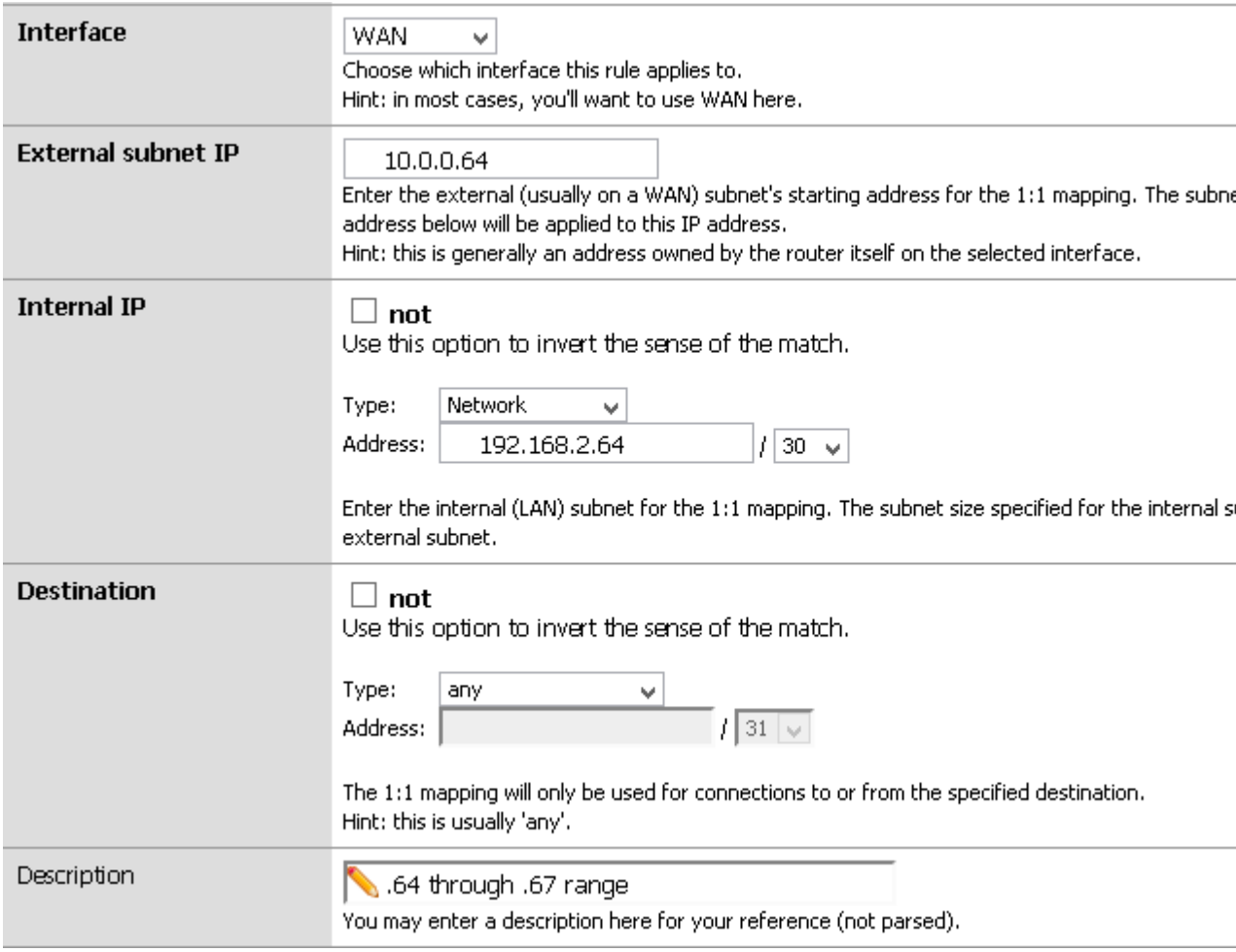
The last octet of the IP addresses need not be the same on the inside and outside, but it's recommended to do so whenever possible. For example, Table 11.2, “/30 CIDR mapping — non-matching final octet” would also be valid.

### Table 11.2. /30 CIDR mapping — non-matching final octet

| External IPs | Internal IPs |
|---|---|
| **10.0.0.64/30** | **192.168.2.200/30** |
| 10.0.0.64 | 192.168.2.200 |
| 10.0.0.65 | 192.168.2.201 |
| 10.0.0.66 | 192.168.2.202 |
| 10.0.0.67 | 192.168.2.203 |

Choosing an addressing scheme where the last octet matches makes your network easier to understand and hence maintain. Figure 11.11, “1:1 NAT entry for /30 CIDR range” shows how to configure 1:1 NAT to achieve the mapping listed in Table 11.1, “/30 CIDR mapping — matching final octet”.

**Figure 11.11. 1:1 NAT entry for /30 CIDR range**




# 1:1 NAT on the WAN IP, aka "DMZ" on Linksys

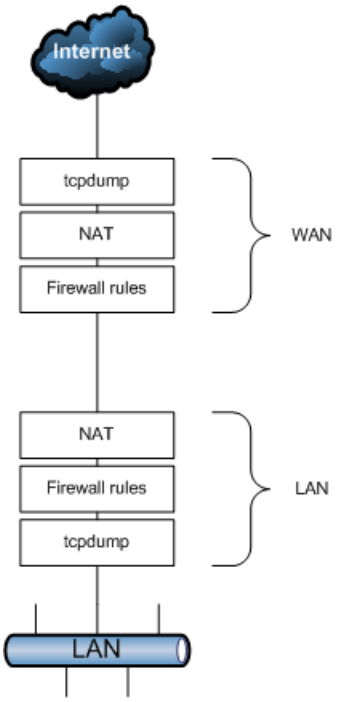
Some consumer routers like those from Linksys have what they call a "DMZ" feature that will forward all ports and protocols destined to the WAN IP address to a system on the LAN. In effect, this is 1:1 NAT between the WAN IP address and the IP address of the internal system. "DMZ" in that context, however, has nothing to do with what an actual DMZ network is in real networking terminology. In fact, it's almost quite 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, you can have 1:1 NAT active on the WAN IP, with the caveat that it will leave all services running on the firewall itself inaccessible externally. So where you are running VPNs of any type, or other local services on the firewall that must be accessible externally, you cannot use 1:1 NAT with your WAN IP. You can work around this with a negated port forward for locally hosted services.

# Ordering of NAT and Firewall Processing

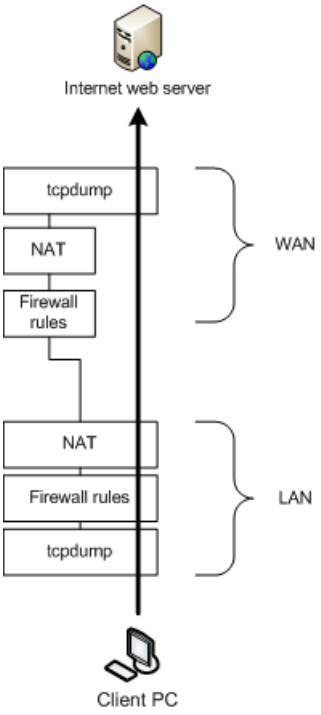
Understanding the order in which firewalling and NAT occurs is important when configuring NAT and firewall rules. The Figure 11.12, “Ordering of NAT and Firewall Processing” illustrates this ordering. It also depicts where **tcpdump** ties in, since its use as a troubleshooting tool will be described later in this book (see Chapter 30, *Packet Capturing*).

**Figure 11.12. Ordering of NAT and Firewall Processing**



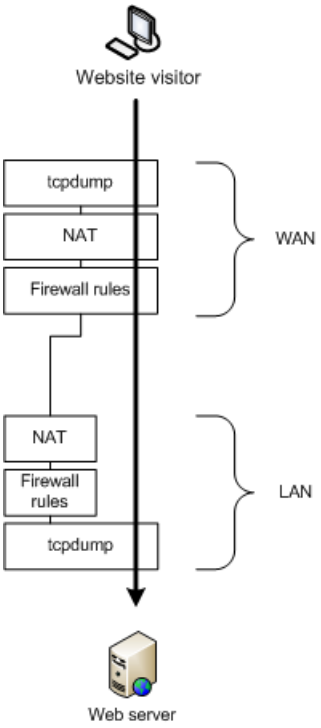

Each layer is not always hit. Figure 11.13, “LAN to WAN Processing” and Figure 11.14, “WAN to LAN Processing” illustrate which layers apply for traffic initiated from the LAN going to the WAN, and also for traffic initiated on the WAN going to LAN (when such traffic is permitted).

**Figure 11.13. LAN to WAN Processing**




For traffic from LAN to WAN, first the firewall rules are evaluated, then the outbound NAT is applied if the traffic is permitted. The WAN NAT and firewall rules do not apply to traffic initiated on the LAN.

**Figure 11.14. WAN to LAN Processing**




For traffic initiated on the WAN, NAT applies first, then the firewall rules.

Note that **tcpdump** is always the first and last thing to see traffic — 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 Chapter 30, *Packet Capturing*)

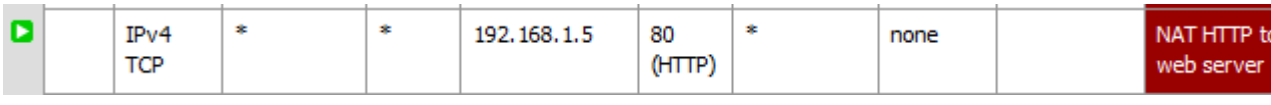
# Extrapolating to additional interfaces

The previous diagrams only illustrate a basic two interface LAN and WAN deployment. When working with firewalls with OPT and OPT WAN interfaces, the same rules apply. All OPT interfaces behave the same as LAN, and all OPT WAN interfaces behave the same as WAN. 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 you define Outbound NAT rules that match traffic between internal interfaces, it will apply as shown.

# Rules for NAT

For rules on WAN or OPT WAN interfaces, because NAT translates the destination IP of the traffic before the firewall rules evaluate it, your WAN firewall rules must always specify the private IP address as the destination. For example, when you add a port forward for TCP port 80 on WAN, and check the Auto-add firewall rule box, this is the resulting firewall rule on WAN. The internal IP on the port forward is 192.168.1.5. Whether using port forwards or 1:1 NAT, firewall rules on all WAN interfaces must use the internal IP as the destination address. Refer to Figure 11.15, “Firewall Rule for Port Forward to LAN Host” for an example of how such a rule should appear.

**Figure 11.15. Firewall Rule for Port Forward to LAN Host**



# NAT Reflection

NAT reflection refers to the ability to access your external services from the internal network by public IP, the same as you would if you were on the Internet. Many commercial and open source firewalls do not support this functionality at all. pfSense has good support for NAT reflection, though some environments will require a split DNS infrastructure to accommodate this functionality. Split DNS is covered in the section called “Split DNS”.

# Configuring and Using NAT Reflection

In pfSense 2.1, the previously confusing NAT reflection wording has been rewritten, partially to account for the recent enhancements to NAT reflection in general.

## NAT Reflection for Port Forwards

To enable NAT reflection, browse to the System → Advanced page. Scroll down under Network Address Translation and select one of the available methods for NAT reflection, as shown in Figure 11.16, “Enable NAT Reflection”. Click Save, and NAT reflection will be enabled. No further configuration is needed, it will immediately work.

**Figure 11.16. Enable NAT Reflection**


There are three available choices for NAT Reflection mode for port forwards, they are:

Disable
: No NAT Reflection will be done by default, but it may be done on a per-rule basis.

Enable (NAT + Proxy)
: The NAT + proxy mode uses a helper program to send packets to the target of the port forward. It is useful in setups where the interface and/or gateway IP used for communication with the target cannot be accurately determined at the time the rules are loaded. Reflection rules are not created for ranges larger than 500 ports and will not be used for more than 1000 ports total between all port forwards. This mode does not work reliably with UDP, only with TCP. Because this is a proxy, the source address of the traffic, as seen by the server, is the firewall's IP address closest to the server.

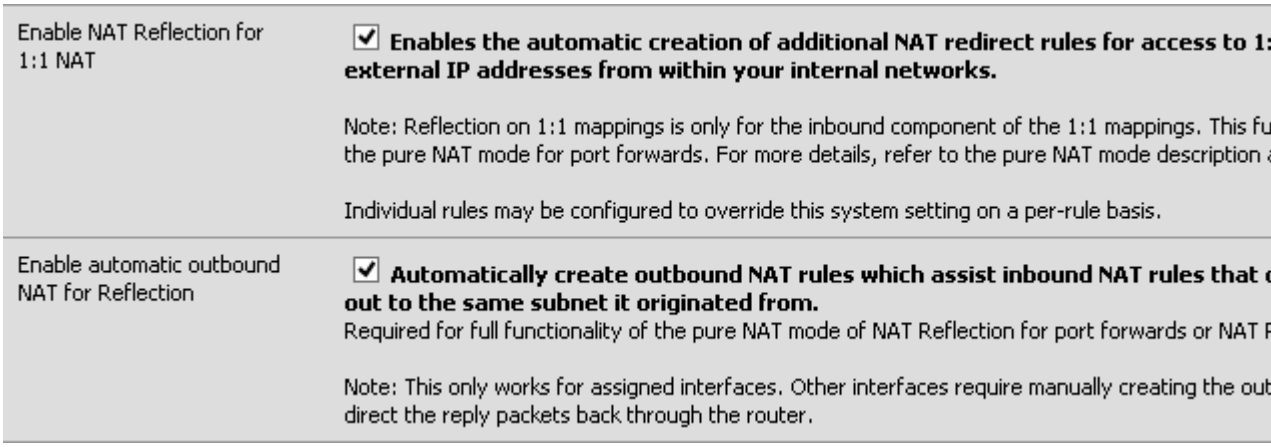
Enable (Pure NAT)
: The pure NAT mode uses a set of NAT rules to direct packets to the target of the port forward. It has better scalability, but it must be possible to accurately determine the interface and gateway IP used for communication with the target at the time the rules are loaded. There are no inherent limits to the number of ports other than the limits of the protocols. All protocols available for port forwards are supported. If you choose this option, and your servers are on the same subnet as your clients, you will also need to check Enable automatic outbound NAT for Reflection a few options down the page from here.

There is also an option for Reflection Timeout that is only used in **`Enable (NAT + Proxy)`** mode. This option controls how long the NAT proxy daemon will wait before closing a connection.

## NAT Reflection for 1:1 NAT

New in pfSense 2.0 is the option to perform NAT reflection for 1:1 NAT entries, allowing your clients on internal networks to reach locally hosted services by connecting to the external IP of a 1:1 NAT entry. This was not possible in pfSense 1.2.x. In order to properly perform 1:1 NAT Reflection for your clients, you should check both Enable NAT Reflection for 1:1 NAT and Enable automatic outbound NAT for Reflection. The latter option is only necessary if your clients and servers are in the same subnet, as it will hide the source address of the client to ensure reply traffic flows back through the firewall so it can be translated back to the original external IP properly.

**Figure 11.17. Enable NAT Reflection for 1:1 NAT**




## NAT Reflection Caveats

NAT reflection is always a bit of a hack as it loops traffic through the firewall. Because of the limited options pf allows for accommodating these scenarios, there are some limitations in the pfSense NAT 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. This has improved somewhat in pfSense 2.1, but still suffers from the limitations imposed by the traffic having to flow through the firewall in both directions. Maintaining a split DNS infrastructure is required by many commercial firewalls even, and typically isn't a problem.

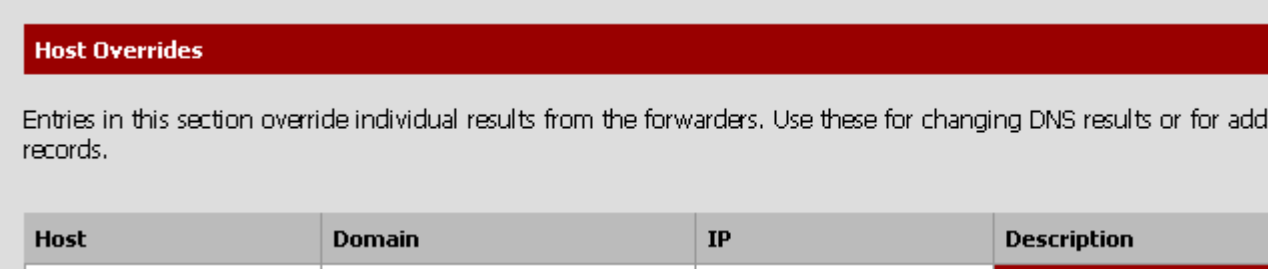
# Split DNS

A preferable alternative to NAT reflection is deploying a split DNS infrastructure. Split DNS refers to a DNS configuration where your public Internet DNS resolves to your public IPs, and DNS on your internal network resolves to the internal, private IPs. The means of accommodating this will vary depending on the specifics of your DNS infrastructure, but the end result is the same. You bypass the need for NAT reflection by resolving hostnames to the private IPs inside your network.

## DNS Forwarder Overrides

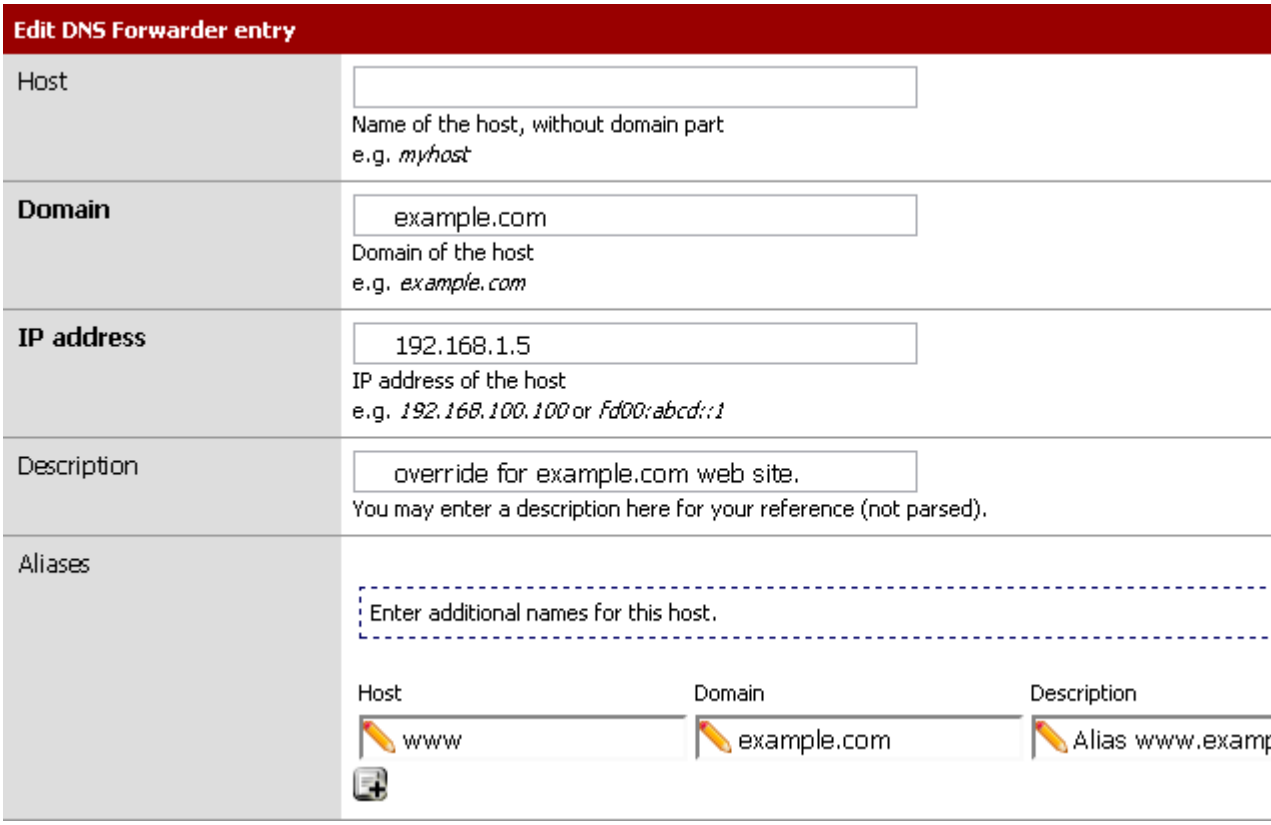
If you use pfSense as your DNS server for internal hosts, you can use DNS forwarder overrides to accomplish a split DNS deployment. To add an override to the DNS forwarder, browse to Services → DNS Forwarder, and click the under "You may enter records that override the results from the forwarders below", as indicated by Figure 11.18, “Add DNS Forwarder Override”.

**Figure 11.18. Add DNS Forwarder Override**




This brings up the DNS forwarder: Edit host screen. The Figure 11.19, “Add DNS Forwarder Override for example.com” shows an example of a DNS override for example.com and www.example.com.

**Figure 11.19. Add DNS Forwarder Override for example.com**




You will need to add an override for each hostname in use behind your firewall.

## Internal DNS servers

If you use other DNS servers on your internal network, such as is common when using Microsoft Active Directory, you will need to create zones for all the domains you host inside your 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. If you are using a different DNS server, it may support similar functionality. Check its documentation for information.

# Outbound NAT

Outbound NAT, also known as "Source NAT" in some firewall software, controls how traffic leaving an interface of your firewall will have its source address and ports translated. To configure it, visit the Firewall → NAT page and choose the Outbound tab. There are two configuration options for Outbound NAT in pfSense, Automatic outbound NAT rule generation and Manual outbound NAT generation (Advanced Outbound NAT (AON)). In networks with a single public IP address per WAN, there is usually no reason to enable AON. In environments with multiple public IP addresses, this may be desirable. For environments using CARP, it is important to NAT outbound traffic to a CARP IP address, as discussed in Chapter 25, *Firewall Redundancy / High Availability*.

As with other rules in pfSense, 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 you are using Manual Outbound NAT.

### Note

Manual Outbound NAT only controls what happens to traffic *as it leaves an interface*. It does *not* control the interface though which traffic will exit the firewall. That is handled by the routing table (the section called “Static Routes”) or policy routing (the section called “Policy routing”).

# Default Outbound NAT Rules

When using the default Automatic outbound NAT, pfSense will automatically create NAT rules translating 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.

If you have no rules in the Outbound NAT list and switch to Manual Outbound NAT, then click Save, a full set of rules will be created for you that are the equivalent of the automatic rules.

# Static Port

By default, pfSense rewrites the source port on all outgoing packets. Some operating systems do a poor job of source port randomization, if they do it at all. This makes IP spoofing easier, and makes it possible to fingerprint hosts behind your firewall from their outbound traffic. Rewriting the source port eliminates these potential (but unlikely) security vulnerabilities.

However, this breaks some applications. There are built in rules when Advanced Outbound NAT is disabled that don't do this for UDP 500 (IKE for VPN traffic) because it will almost always be broken by rewriting the source port. All other traffic has the source port rewritten by default.

You may use other protocols, like some games amongst other things, which do not work properly when the source port gets rewritten. To disable this functionality, you need to use the static port option. Click Firewall → NAT, and the Outbound tab. Click Manual Outbound NAT rule generation (Advanced Outbound NAT (AON)) and click Save. You will then see a rule at the bottom of the page labeled Auto created rule for LAN. Click the button to the right of that rule to edit it. Check the Static Port box on that page, and click Save. Apply Changes. After making that change, the source port on outgoing traffic will be preserved. You can also do this more selectively by adding a rule at the top of the list to match only a specific device, such as a PBX or a game console, and do static port NAT for just that single device. It is better to do this in a more selective fashion, to avoid any potential conflict if two local hosts use the same source port to talk to the same remote server and port.

# Disabling Outbound NAT

If you are using public IP addresses on local interfaces, and thus do not need to apply NAT to traffic passing through the firewall, you should disable NAT for that interface. In order to do this, you must

first change the Outbound NAT setting to Manual Outbound NAT, and then Save. After making that change, one or more rules will appear in the list on the Outbound NAT screen. Delete the rule or rules specifying the source of the public IP subnets by clicking each line once (or check the box at the start of the line) and then click the button at the bottom of the list. Click Apply Changes to complete the process.

Once all of the rules have been deleted, outbound NAT will no longer be active for those source IP addresses, and pfSense will then route public IP addresses without translation.

To completely disable outbound NAT, delete all of the rules that are present when using Manual Outbound NAT.

# Working with Manual Outbound NAT Rules

Manual Outbound NAT rules are very flexible and are capable of translating your traffic in many ways. Outbound NAT rules are managed and processed like many other rule types in pfSense. The rules are matched from the top-down, and the first match is used. Because the NAT rules are shown in a single page the Interface column is a source of confusion for some; As traffic leaves an interface, only the outbound NAT rules for that interface are consulted.

## Note

In versions of pfSense prior to 2.1, the button at the top of the list added the rule to the bottom of the list. This has now been changed so that the button at the top of the list adds a rule to the top of the outbound NAT rules.

The options for each Outbound NAT rule are:

| | |
|---|---|
| Do not NAT | Checking this option will cause packets matching the rule to *not* have NAT applied as they leave. This would only be needed if the traffic would otherwise match a NAT rule, but should not have NAT. One common use for this is to add a rule exception so that the firewall's IPs 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 | This is 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 other internal interface. |
| Protocol | Most of the time you will want to apply Outbound NAT to **any** protocol, but in certain cases you may want to restrict the protocol upon which the NAT will act. For example, you may want to only do static port NAT out for UDP traffic from your PBX. |
| Source | The Source is the local network which will have its address translated as it leaves Interface. This is typically your LAN, DMZ, or a VPN subnet. The Source Port should almost always be left blank to indicate any. |
| Destination | In most cases, the Destination is left set to **any** so that traffic going anywhere out of this Interface will be translated, but you can restrict the Destination however you like if you only need to translate in a certain way when going to a specific destination. For example, only doing static port NAT to your SIP trunks. An alias may be used in this field if the Type is set to **Network**. |
| Translation | The Address field inside of the Translation section controls exactly what happens to the source address of the traffic when this rule is matched. Most commonly, this is set to **Interface Address** so the traffic is translated to the IP address of Interface, e.g. your WAN IP address. The Address drop-down also contains all of your Virtual IPs, 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, you can translate to a pool of addresses. This can help in large NAT deployments or in areas where you need static port for several clients. When translating to a host alias or subnet, you will also see a Pool Options drop-down 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. |
| Bitmask | Applies the subnet mask and keeps the last portion identical. For example if the source address was 10.10.10.50 and the translation subnet was 192.2.0.0/24, it would be translated to 192.2.0.50. This works similarly to 1:1 NAT but only in the outbound direction. |

The Port field lets you specify a specific source port for translation. This is most always left blank, but could be required if the client selects a random source port but the server requires a specific source port.

Checking the Static Port option will cause the original source port of the client's 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.

No XMLRPC Sync — If you are not using a CARP failover cluster, skip over the No XML-RPC Sync option. If you are, then checking this box will prevent this rule from being synchronized to the other members of a failover cluster (see Chapter 25, *Firewall Redundancy / High Availability*), which is usually undesirable. 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.

Using these rules, you can accommodate most any NAT scenario, large or small.

## Tracking Changes to Outbound NAT Rules

As mentioned in Figure 10.5, “Firewall Rule Time Stamps” for firewall rules, a timestamp is added to an outbound NAT entry when it is created or last edited, to show 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

Your choice of NAT configuration will depend primarily on the number of public IPs you have and number of systems that require inbound access from the Internet.

## Single Public IP per WAN

When you have only a single public IP per WAN, your NAT options are limited. You can use 1:1 NAT with WAN IPs, but that can have drawbacks. In this case, it is advisable to only use port forwards.

## Multiple Public IPs per WAN

With multiple public IPs per WAN, you have numerous options for your inbound and outbound NAT configuration. Port forwards, 1:1 NAT, and Advanced Outbound NAT may all be desirable in some circumstances.

# NAT and Protocol Compatibility

Some protocols do not work well and some not at all with NAT. Some protocols embed IP addresses within packets, some do not work properly if the source port is rewritten, and some are difficult because of limitations of pf. This section covers the 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 does some things that are very NAT and firewall unfriendly. pfSense now uses an in-kernel FTP proxy that can handle more flexible situations than the previous pair of userland FTP proxy programs.

### FTP Limitations

Because pf lacks the ability to properly handle FTP traffic without a proxy, and the pfSense FTP proxy implementation is somewhat lacking, there are some restrictions on the usage of FTP.

#### FTP servers behind NAT

FTP servers behind NAT must use port 21 by default, as the FTP proxy will only launch when port 21 is specified. You can configure the FTP proxy to attach on other ports by setting the system tunable `debug.pfftpports` to another value. By default it is simply *21*, but if you set it to "*21 2121*" it

would be active on both ports *21* and *2121*. See the section called “System Tunables Tab” for more on setting system tunables.

# 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 you are hosting a server or acting as a client.

## 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. As you may have guessed, 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 and a port forward allowing traffic into this port.

When the FTP proxy is in use, it 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.

When everything is working as it should, this all happens transparently. The server never knows it's 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, this 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.

## 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 have to give 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.

## 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

To ensure the FTP proxy works properly for port forwards, a port forward NAT rule should be set for exactly the FTP port, e.g. 21, and not a range or alias containing multiple ports.

# FTP Servers and 1:1 NAT

Unlike versions pfSense before 2.x, nothing special is required for an FTP server on a 1:1 NAT entry any longer.

# TFTP

Standard TCP and UDP traffic initiate connections to remote hosts using a random source port in the ephemeral port range (range 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's destination port, and the destination port is the client's source port. This is how pf associates the reply traffic with connections initiated from inside your network.

TFTP (Trivial File Transfer Protocol) does not follow this, 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. Your 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 you must pass TFTP through the firewall, there is a TFTP proxy that is configurable under System → Advanced on the Firewall/NAT tab. See the section called “TFTP Proxy” for more information.

# PPTP / GRE

The limitations with PPTP in pfSense are caused by limitations in pf's ability to NAT the GRE protocol. As such, the limitations apply to any use of the GRE protocol, however PPTP is the most common use of GRE in most networks today.

The state tracking code in pf for the GRE protocol can only track a single session per public IP per external server. This means if you use PPTP VPN connections, only one internal machine can connect simultaneously to 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 IPs on your firewall, one per client via Outbound or 1:1 NAT, or to use multiple public IPs on the external PPTP server. This is not a problem with other types of VPN connections.

Due to the same GRE limitations mentioned above, if you enable the PPTP Server on pfSense, you cannot connect to any PPTP server on the Internet from clients NATed to the WAN IP on pfSense. The work around for this also requires the use of more than one public IP address. You can NAT internal clients to another public IP, and only be subject to the same per-public IP restrictions mentioned previously.

Since we largely rely on the functionality of the underlying system, this is a difficult problem for us to solve. We investigated potential solutions for this in pfSense 2.x, but never found a stable solution. Due to PPTP's extremely flawed security (the section called “PPTP Security Warning”), including a complete compromise of the entire protocol, its usage should be discontinued as soon as possible, so this issue is not quite as relevant given the current circumstances.

# Online Games

Games typically are NAT friendly aside from a couple caveats. This section refers to PC games with online capabilities as well as console gaming systems with online capabilities. This section provides an overview of the experiences of numerous pfSense users. I recommend visiting the Gaming board on the pfSense forum [http://forum.pfsense.org] to find more information.

## Static Port

Some games do not work properly unless you enable static port. If you are having problems with a game, the best thing to try first is enabling static port. See the static port section earlier in this chapter 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 you are using and you are likely 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 special needs in terms of NAT port forwards and firewall rules. You may find that enabling UPnP on your pfSense system will easily allow games to work with little or no intervention. See the section called “UPnP & NAT-PMP” for more information on configuring and using UPnP.

# IPv6 Network Prefix Translation (NPt)

Network Prefix Translation or NPt for short, works similarly to 1:1 NAT. NPt can be found under Firewall → NAT on the NPt tab. NPt will take one prefix and translate it to another. So you could translate *2001:db8:1111:2222::/64* to be *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, you can use a "private" IPv6 space (fc00::/7) on your LAN and have it translated to one of your public, routed, IPv6 prefixes as it comes and goes from your LAN. The utility of this is debatable. One use is if you plan on changing external providers but do not want to renumber your LAN, however since you'd have to also adjust anything external that looked for the old prefix, the usefulness of that can go either way, especially when you consider that you would have to account for avoiding collisions in the fc00::/7 space for VPN tunnels as well.

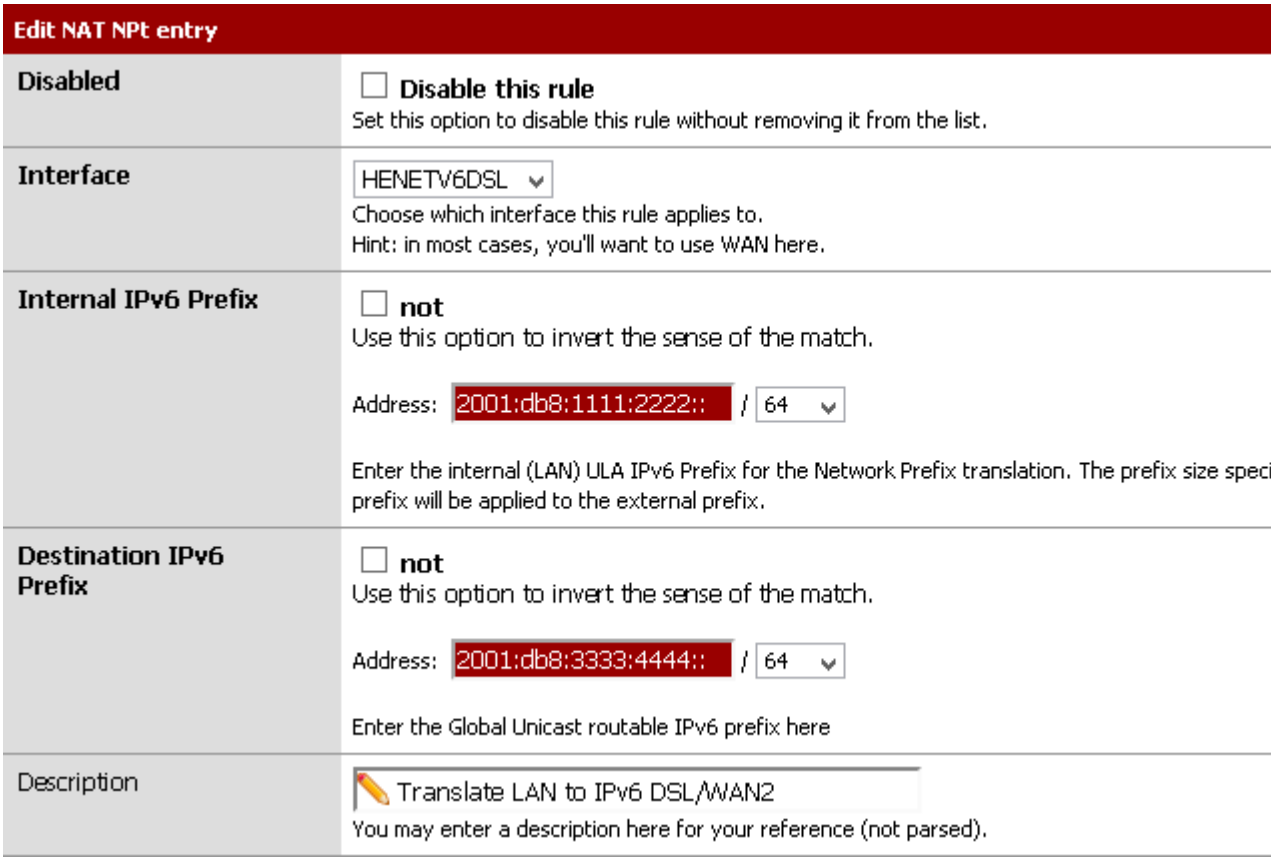
NPt makes perfect sense for SOHO IPv6 Multi-WAN setups. The likelihood that an end user will have their own provider-independent IPv6 space and a BGP feed is very small. In these cases, you can utilize a routed prefix from multiple WANs to function similarly to Multi-WAN on IPv4. As traffic leaves the second WAN, if it's coming from the LAN subnet, it will be translated to the equivalent IP in that WAN's routed subnet. You can either use one of your routed prefixes on LAN and do NPt on the other WANs, or use addresses in fc00::/7 on LAN and do NPt on both WANs. Personally I would avoid using the fc00::/7 space for this task. For more information on Multi-WAN with IPv6, see the section called “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** of your 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. |
| Description | A brief description of the purpose for this entry. |

In Figure 11.20, “NPt Example”, the LAN IPv6 subnet *2001:db8:1111:2222::/64* will be translated to *2001:db8:3333:4444::/64* as it leaves the HENETV6DSL interface.

**Figure 11.20. NPt Example**




# Troubleshooting

NAT can be a complex animal, and in all but the most basic environments, there are bound to be some issues getting a good working configuration. This section will go over a few common problems and some suggestions on how they might be solved.

# 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 our users have been solved by one or more of the following suggestions.

## Port forward entry incorrect

Before any other troubleshooting task, ensure that your settings for the port forward are correct. Go over the process in the section called “Adding Port Forwards” again, and double check that the values are correct. Remember, if you change the NAT IP or the Ports, you will also need to adjust the matching firewall rule if you are not using linked firewall rules. Common things to check for:

- Correct interface (usually should be WAN, or wherever traffic will be entering the pfSense box).
- Correct NAT IP, which must be reachable from an interface on the pfSense router.
- Correct port range, which must correspond to the service you are trying to forward.
- 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 (the section called “Viewing the Firewall Logs”). Remember, that the destination for the firewall rule should be the *internal* IP address of the *target system* and not the address of the interface containing the port forward. See the section called “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, you will need to check its logs and settings to confirm whether or not the traffic is being blocked at that point.

## pfSense is not the target system's gateway

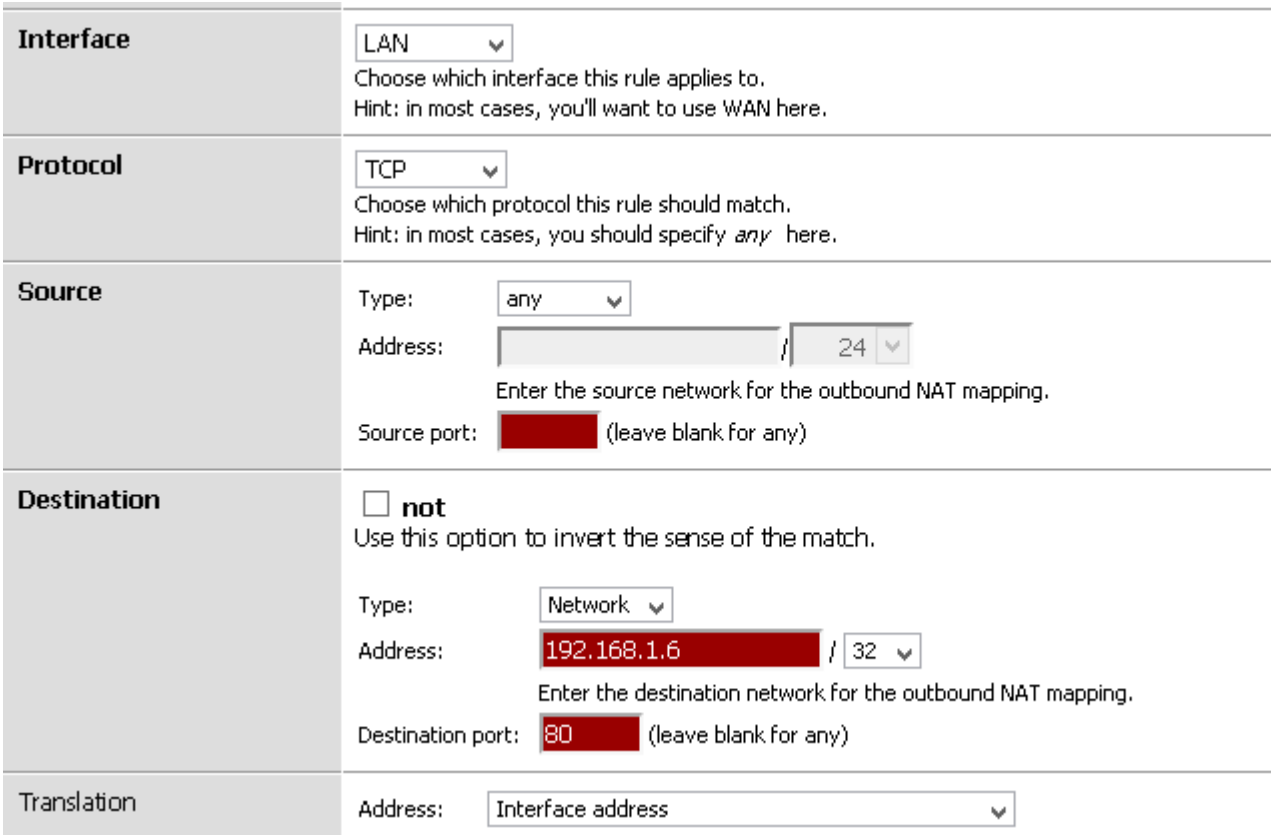
In order for pfSense to properly forward a port for a local system, pfSense must be the default gateway for the target system. If pfSense is not the gateway, the target system will attempt to send replies to port forward traffic out whatever system *is* the gateway, and then one of two things will happen: 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, cannot have a gateway, or cannot use pfSense as its gateway

A subset of the larger problem of the target's machine's gateway is when your device has no gateway, or is incapable of having a gateway. In these cases, you can work around that problem by switching to 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 that system on the target port.

For example, if you have an IP camera that does not support a gateway located at 192.168.1.6, you can switch to manual outbound NAT and create a rule like Figure 11.21, “Manual Outbound NAT rule for LAN device with missing gateway” to reach it from outside the network. The IP camera will see the LAN IP address of the firewall as the source of the traffic, and since that is "local" to the camera, it will respond properly.

**Figure 11.21. Manual Outbound NAT rule for LAN device with missing gateway**




## Target machine is not listening on the forwarded port

If, when the connection is tested, the request is rejected instead of timing out, 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 be listening for SSH connections, but the port forward was entered for port 23 instead of 22, the request would most likely be rejected. You can usually tell the difference by trying to connect to the port in question using **telnet**. A message such as **Connection refused** indicates something, frequently the inside host, is actively refusing the connection. Using Diagnostics → Test Port can also help, see the section called “Testing a TCP Port”.

## ISP is blocking the port you are trying to forward

In some cases, ISPs will filter incoming traffic to well-known ports. Check your ISP's Terms of Service (ToS), and 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 your ISP, you may need to move your services to a different port in order to work around the filtering. For example, if your ISP disallows servers on port 80, try 8080 or 8888.

Before attempting to work around a filter, consult your ISP's ToS to ensure you are not violating their rules.

## Testing from inside your network instead of outside

By default, port forwards will only work when connections are made from outside of your network. This is a very common mistake when testing port forwards.

If you require port forwards to work internally, see the section called “NAT Reflection”. However, Split DNS (the section called “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 your time to implement that instead. Even with NAT reflection, testing from inside your network isn't necessarily indicative of whether it will work from the Internet. ISP restrictions, restrictions on devices upstream of your firewall, amongst other possibilities are not possible to see when testing from within your network.

## Incorrect or missing Virtual IP address

When using IP addresses that are not the actual IP addresses assigned to an interface, you must use Virtual IPs (VIPs, see the section called “Virtual IPs”). If a port forward on an alternate IP address is not working, you may need to switch to a different type of VIP. For example, you may need to use a Proxy ARP type instead of an "Other" type VIP.

When testing, also make sure that you are 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 would need to 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 the section called “Port forwards to hosts behind the portal only work when the target system is logged into the portal” for the details on how to get that to work.

## Further testing needed

If none of these solutions helped you obtain a working port forward, consult the section called “Firewall States” to look for NAT states indicating that the connection has made it through the firewall, or Chapter 30, *Packet Capturing* for information on using packet captures to diagnose port forwarding issues.

# NAT Reflection Troubleshooting

NAT Reflection (the section called “NAT Reflection”) is complex, and as such may not work in some advanced scenarios. We recommend that you use Split DNS instead (see the section called “Split DNS”) in most cases. However, NAT Reflection in 2.0.3, 2.1 and newer releases works well for nearly all scenarios, and any problems are usually a configuration mistake. Ensure that it was enabled the right way, and make sure you are not forwarding a large range of ports.

NAT Reflection rules are also duplicated for each interface present in the system, so if you have a lot of port forwards and interfaces, the number of reflectors can easily surpass the limits of the system. If this happens, an entry is printed in the system logs. Make sure to check the system logs for any errors or information.

## Web Access is Broken with NAT Reflection Enabled

If you have an improperly specified NAT Port Forward, it can cause problems when NAT Reflection is enabled. The most common way this problem arises is when you have 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 you make comes up as your own web site. To fix this, edit your NAT Port Forward for the offending port, and change External Address to **Interface Address** instead.

If you really require an external address of **any**, then NAT Reflection will not work for you, and you'll need to employ Split DNS instead.

# Outbound NAT Troubleshooting

When you have manual outbound NAT enabled, and there are multiple local subnets, an outbound NAT entry will be needed for each. This applies especially if you intend to have traffic exit with NAT after coming into the pfSense router via a VPN connection such as PPTP or 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 Chapter 30, *Packet Capturing* for more details on obtaining and interpreting packet captures.

# Chapter 12. Routing

One of the primary functions of a firewall is routing traffic, in addition to filtering and performing NAT. This chapter covers several routing related topics, including gateways, static routes, routing protocols, routing of public IPs, and displaying routing information.

# Gateways

The key to routing is gateways, or 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 reach 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 you have an IP address of `192.168.22.5` on a firewall, 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 PPP-based interfaces which often have gateway IPs 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, to route an IPv6 network, you must do so via an IPv6 gateway/router. You cannot create a route 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

Whether you need to add a default gateway, an additional gateway for a static route, or another gateway for Multi-WAN, the gateways must be added before they may be used. If you are adding a gateway for a WAN-type interface, you can do so from that interface (See the section called “Interface Configuration Basics”), or add the gateway here and then select it 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, on the Gateways tab. To add a new gateway, click the button at the top or bottom of the list. To edit an existing gateway, click the button at the end of its row. To delete a gateway, click the button at the end of its row.

## Gateway Settings

When adding or editing a gateway, you will be presented with a screen that lists all of the options for controlling a gateway's behavior. The only required settings are the Interface, the Name, and the Gateway (IP address).

### Interface

This is the interface through which the gateway is reached. For example, if this is a local gateway on your LAN subnet, you would choose the LAN interface here.

## 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. So you could have *WANGW*, *GW_WAN*, or *WANGATE*, but not *WAN GW*.

## Gateway

This is the IP address of the gateway. As mentioned previously, this should reside in the same subnet as the chosen interface.

## Default Gateway

This checkbox controls whether 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. You may 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 the gateway's status in terms of latency and packet loss. This data is used for the gateway status information and also to draw the Quality RRD graph. If you find this monitoring 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 such failures.

## Monitor IP

The Monitor IP option lets you choose the IP address to ping for monitoring the gateway's status. By default the system will ping the gateway IP address for monitoring the status. This is not always desirable, especially in the case where the gateway IP is local to you, 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 having upstream failures, so pinging a host on the Internet is a more accurate test of the WAN's usability rather than testing the link itself. Some popular choices include OpenDNS servers, Google's 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 goes out the expected gateway. Each gateway must have a unique Monitor IP. You can verify if a gateway is perceived as online by the firewall by checking Status → Gateways or by using the Gateways widget on the dashboard. If it shows Online, then the monitor IP is successfully returning pings.

## Advanced

There are several parameters that can be changed which affect 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 Advanced button. If any of the advanced options are set, this section is automatically expanded. For more information on using multiple WAN connections, see Chapter 15, *Multiple WAN Connections*.

### Weight

When using Multi-WAN, if two WANs have different amounts of bandwidth, the Weight parameter can be used to adjust how they are used. For example if you have 5Mbit/s WAN and 10Mbit/s WAN2, you can weight WAN as 1 and WAN2 as 2. Then for every three connections that go out, one will use WAN and two will use WAN2. This will more accurately distribute the bandwidth on this type of setup.

### 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. 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 working fine. Another example is a GIF tunnel to a place like 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.

### Packet Loss Thresholds

Similar to the Latency Thresholds above, the Packet Loss Thresholds control the amount of packet loss that the system can see to a monitor IP 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. 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 drops or delays ICMP in favor of other traffic. We have seen connections be unusable with minor amounts of loss, and some that are usable even when showing 45% loss. If you find that you are seeing loss alarms on a normally functioning WAN gateway, enter higher values in the From and To fields until you achieve a good balance for that circuit.

### Probe Interval

The value in the Probe Interval field controls how often a ping is sent to the monitor IP. The default is to ping every second. 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 the value in the Down field.

Keep in mind that the quality graph is averaged over seconds, not intervals, so as the Probe Interval is increased the accuracy of the quality graph is decreased.

### Down

The Down field specifies how many seconds must pass while a gateway is in an abnormal state before it is considered down and removed from service. By default this is *10* seconds. Some fine tuning may be needed to prevent false positives (or false negatives), but generally speaking the default value is sufficient.

The Down time defines the length of time before the gateway is marked as down, but the accuracy is controlled by the Probe Interval. For example, if your Down time is 40 seconds but on a 30 second Probe Interval, only one probe would have to fail before the gateway is marked down at the 40 second mark. By default, the gateway is considered Down after 10 seconds and the Probe Interval is 1 second, so 10 probes would have to fail before the gateway is marked down.

## Description

The gateway's Description field is for your reference. A short note about the gateway or interface it's used for may be helpful, or it can be left blank if desired.

# Gateway Groups

Gateway Groups define sets of gateways to be used for failover or load balancing. For information on setting up those features, see Chapter 15, *Multiple WAN Connections*.

# Static Routes

Static routes are used when you have hosts or networks reachable via a router other than your default gateway. Your firewall or router knows about the networks directly attached to it, and reaches all other networks as directed by its routing table. In networks where you have an internal router connecting additional internal subnets, you must define a static route for that network to be reachable. The routers through which these other networks are reached must first be added as gateways. See the section called "Gateways" for information on adding gateways. Static routes are found under System → Routing on the Routes tab.

## Example static route

Figure 12.1, "Static Route" illustrates a scenario where a static route is required.

**Figure 12.1. Static Route**


Because the 192.168.2.0/24 network in Figure 12.1, "Static Route" is not on a directly connected interface of pfSense, you need a static route so it knows how to reach that network. Figure 12.2, "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.

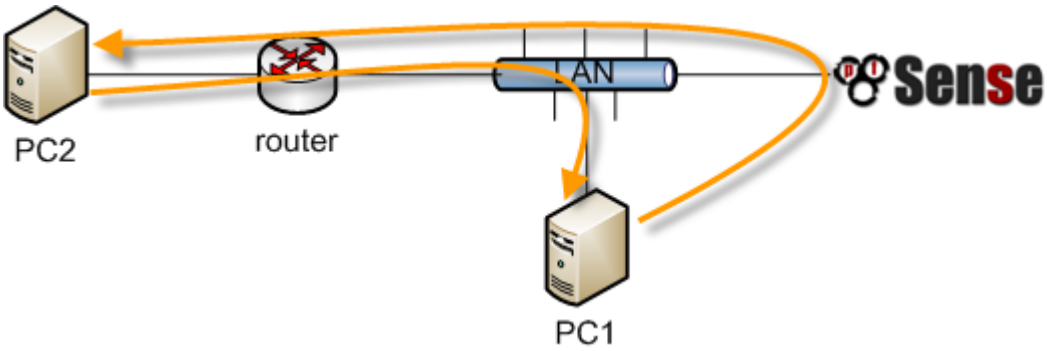
**Figure 12.2. Static route configuration**


The Destination network specifies the subnet reachable via this route. Gateway selects the a router through which this network is reachable. Firewall rule adjustments may also be required. The default LAN rule only allows traffic sourced from the LAN subnet, so if you maintained that rule, you will have to open up the source network to also include the networks reachable via static routes on LAN. The next section describes a common scenario with static routes that you also should review.

# Bypass Firewall Rules for Traffic on Same Interface

In many situations when using static routes you end up with asymmetric routing. This means the traffic in one direction will take a different path from the traffic in the opposite direction. Take Figure 12.3, “Asymmetric routing” for example.

**Figure 12.3. Asymmetric routing**




Traffic from PC1 to PC2 will go through pfSense since it is PC1's default gateway, but traffic in the opposite direction will go directly from the router to PC1. Since pfSense is a stateful firewall, it must see all of the connection to be able to filter traffic properly. With asymmetric routing like this, any stateful firewall will end up dropping legitimate traffic because it cannot properly keep state without seeing traffic in both directions. Always check the Bypass firewall rules for traffic on the same interface box on the System → Advanced page under the Firewall/NAT tab in asymmetric routing scenarios to prevent legitimate traffic from being dropped. This adds firewall rules allowing all traffic between networks defined in static routes using PF's no state option. Alternatively, you can add firewall rules yourself specifying **none** as the State Type, matching traffic between the local and remote subnets, but that is usually not recommended due to the complexity it can introduce and the increased likelihood of mistakes. Should you need to filter traffic between statically routed subnets, 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 added to the routing table of the sending device, and the device will subsequently use that more direct route to reach that network. This will not work if your OS is configured to not permit ICMP redirects, which is typically not the case by default.

ICMP redirects are common when you have a static route pointing 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 mostly undeservedly gotten a bad reputation from some in the security community because they allow modification of a system's 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 the original datagram's data. 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 IPs

This section covers the routing of public IPs, where you have a public IP subnet assigned to an internal interface, and single firewall deployments. If you are using CARP, see the section called “Providing Redundancy Without NAT”.

# IP Assignments

You need at least two public IP subnets assigned to you by your ISP. One is for the WAN of your 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 12.1, “WAN IP Block” and a /29 public subnet on an internal OPT interface as shown in Table 12.2, “Inside IP Block”.

### Table 12.1. WAN IP Block

| 11.50.75.64/30 | |
|---|---|
| **IP Address** | **Assigned To** |
| 11.50.75.65 | ISP router (pfSense's default gateway IP) |
| 11.50.75.66 | pfSense WAN interface IP |

### Table 12.2. 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 | |

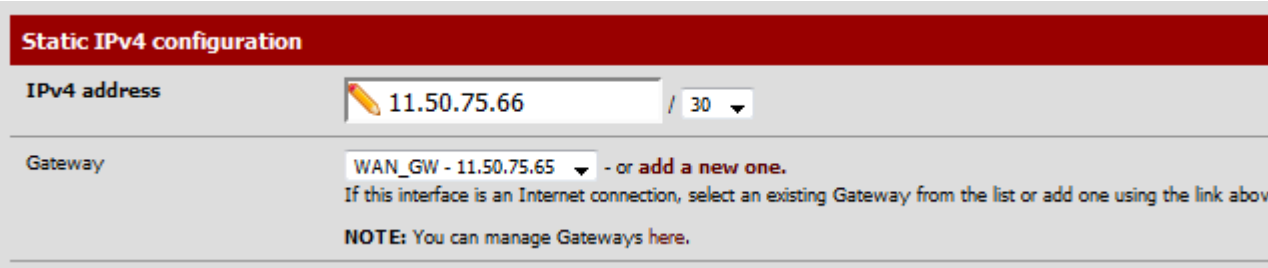
# Interface Configuration

First configure the WAN and OPT interfaces. The LAN interface can also be used for public IPs if you desire. 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 12.4, “WAN IP and gateway configuration” shows the WAN configured as shown in Table 12.1, “WAN IP Block”.

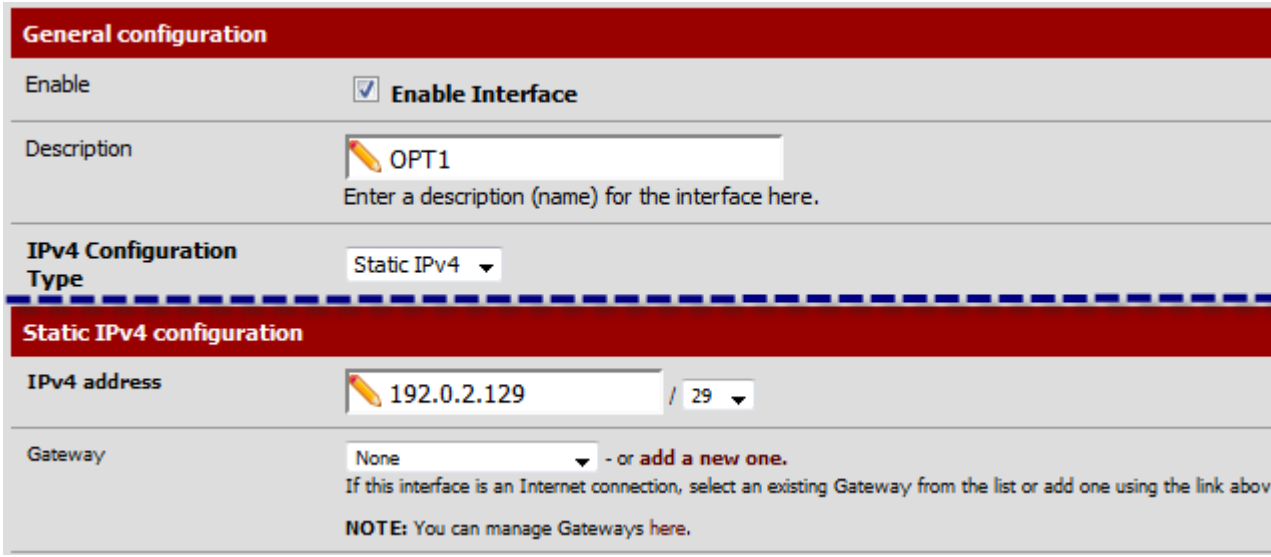
### Figure 12.4. WAN IP and gateway configuration




## Configure OPT1

Now enable OPT1, optionally change its name, and configure the IP address and mask. Figure 12.5, “Routing OPT1 configuration” shows OPT1 configured as shown in Table 12.2, “Inside IP Block”.

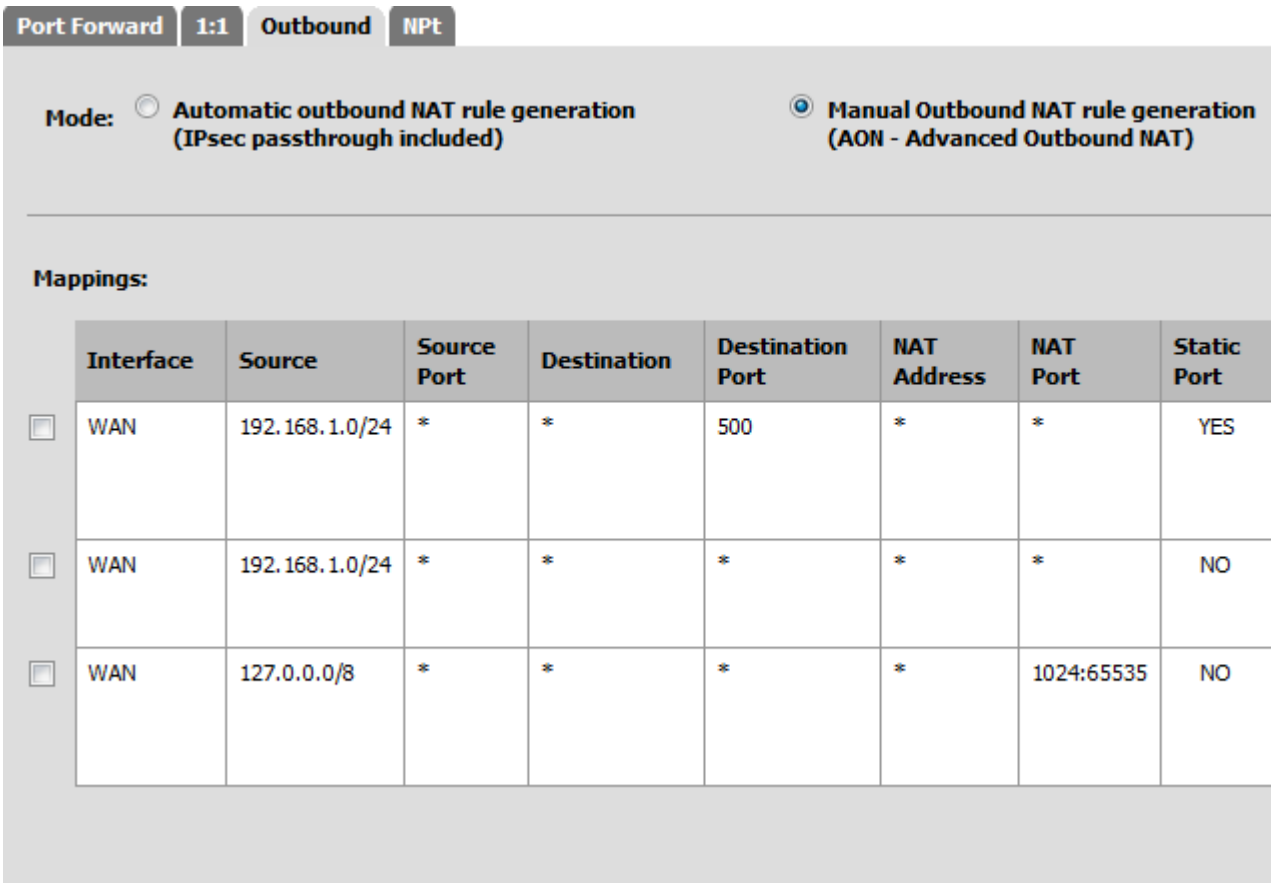
**Figure 12.5. Routing OPT1 configuration**




# NAT Configuration

The default of translating internal traffic to the WAN IP must be overridden when using public IPs on an internal interface. Browse to Firewall → NAT, and click the Outbound tab. Select Manual Outbound NAT rule generation and click Save. This will generate a default rule translating all traffic from the LAN subnet leaving the WAN interface to the WAN IP, the default behavior of pfSense. If your LAN contains a private subnet as in this example, this is the exact desired configuration. Traffic sourced from the OPT1 network's 192.0.2.128/29 is not translated because the source is limited to 192.168.1.0/24. This configuration is shown in Figure 12.6, “Outbound NAT configuration”. If you use public IPs on your LAN, delete this automatically added entry. Then click Apply Changes.

**Figure 12.6. Outbound NAT configuration**

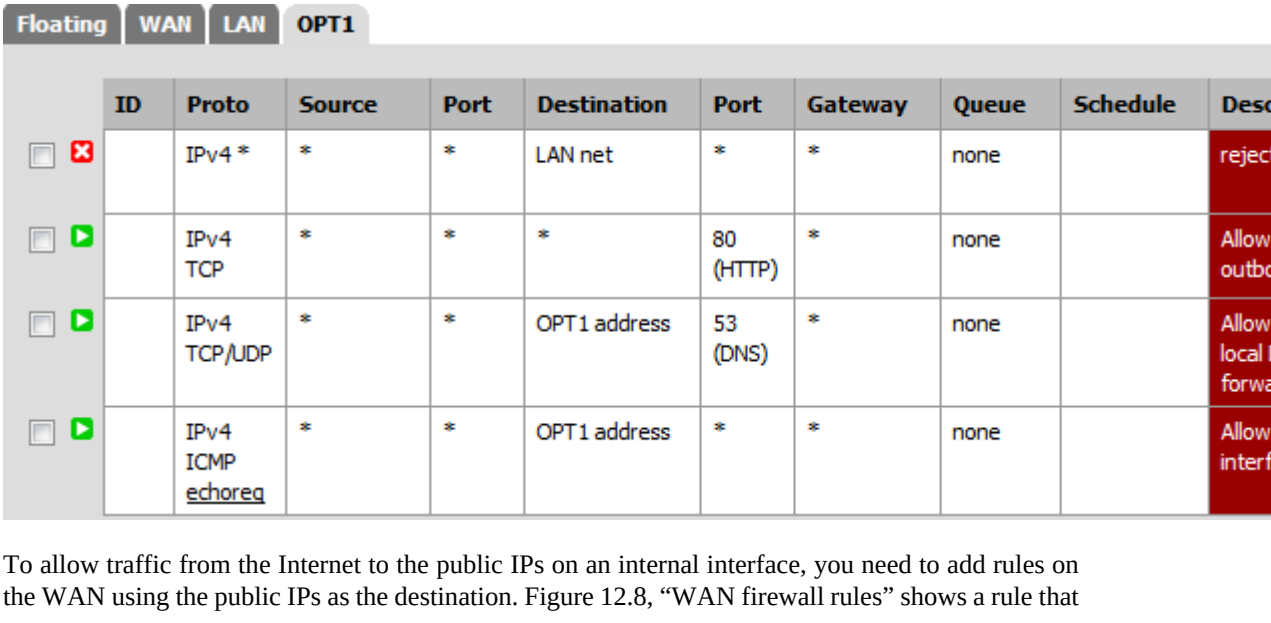



| Interface | Source | Source Port | Destination | Destination Port | NAT Address | NAT Port | Static Port |
|---|---|---|---|---|---|---|---|
| WAN | 192.168.1.0/24 | * | * | 500 | * | * | YES |
| WAN | 192.168.1.0/24 | * | * | * | * | * | NO |
| WAN | 127.0.0.0/8 | * | * | * | * | 1024:65535 | NO |

# Firewall Rule Configuration

The NAT and IP configuration is now complete. Firewall rules will need to be added to permit outbound and inbound traffic. Figure 12.7, “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 are permitted, and HTTP is allowed outbound.

**Figure 12.7. OPT1 firewall rules**

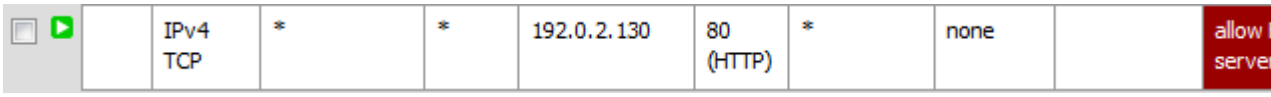



| ID | Proto | Source | Port | Destination | Port | Gateway | Queue | Schedule | Desc |
|---|---|---|---|---|---|---|---|---|---|
| | IPv4 * | * | * | LAN net | * | * | none | | reject |
| | IPv4 TCP | * | * | * | 80 (HTTP) | * | none | | Allow outbo |
| | IPv4 TCP/UDP | * | * | OPT1 address | 53 (DNS) | * | none | | Allow local forwa |
| | IPv4 ICMP echoreq | * | * | OPT1 address | * | * | none | | Allow interf |

To allow traffic from the Internet to the public IPs on an internal interface, you need to add rules on the WAN using the public IPs as the destination. Figure 12.8, “WAN firewall rules” shows a rule that

allows HTTP to 192.0.2.130, one of the public IPs on the internal interface as shown in Table 12.2, “Inside IP Block”.

**Figure 12.8. WAN firewall rules**




After configuring the firewall rules as desired, your setup is complete.

# Routing Protocols

At the time of this writing, three routing protocols are supported with pfSense, RIP (Routing Information Protocol), BGP (Border Gateway Protocol), and 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, visit System → Packages, and click the plus to the right of routed. Once installed, it can be configured under Services → RIP. To use it:

1. Check the Enable RIP box
2. Choose the interfaces RIP will listen and send routing updates on
3. Select your RIP version
4. When using RIPv2, enter a RIPv2 password if one is used on your network.
5. Click Save

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

## BGP

A BGP package using OpenBSD's OpenBGPD [http://www.openbgpd.org] is available. To install it, visit System → Packages, and click the plus to the right of OpenBGPD. Click OK to install the package. You will find OpenBGPD under the Services menu.

BGP is a complex beast, and describing it in detail is outside the scope of this book. Configuration of pfSense's OpenBGPD is straight forward if you understand BGP. During development of this package, we relied on *O'Reilly's BGP* book [http://www.amazon.com/gp/product/0596002548?ie=UTF8&tag=pfsense-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0596002548] and recommend it for anyone looking to deploy BGP.

## OSPF

An OSPF package using the Quagga [http://www.nongnu.org/quagga/] routing daemon is also available. As with BGP, to install it visit System → Packages, and click the plus to the right of Quagga OSPF. Click OK to install the package. You will find Quagga OSPFd under the Services menu.

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 should be familiar. The options for the Quagga OSPF package are similar to those for the old OpenOSPFD package in previous versions of pfSense,

however we have found that the Quagga version is much more well-behaved when operating with other OSPF-enabled equipment.

# Route Troubleshooting

When diagnosing traffic flow issues, one of the first thing 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, visit Diagnostics → Routes and you will see the output like that shown in Figure 12.9, “Route Display”.

**Figure 12.9. Route Display**

**IPv4**

| Destination | Gateway | Flags | Refs | Use | Mtu | Netif |
|---|---|---|---|---|---|---|
| default | 10.0.2.2 | UGS | 0 | 53 | 1500 | le0 |
| 10.0.2.0/24 | link#1 | UC | 0 | 0 | 1500 | le0 |
| 10.0.2.2 | 52:54:00:12:35:02 | UHLW | 2 | 30 | 1500 | le0 |
| 10.0.2.15 | 127.0.0.1 | UGHS | 0 | 0 | 16384 | lo0 |
| 127.0.0.1 | 127.0.0.1 | UH | 1 | 0 | 16384 | lo0 |
| 192.168.56.0/24 | link#2 | UC | 0 | 0 | 1500 | le1 |
| 192.168.56.101 | 08:00:27:00:d4:84 | UHLW | 1 | 521 | 1500 | le1 |

The output from the command line is similar to that seen in the WebGUI:

```
# netstat -rnW
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif Expire
default            10.0.2.2           UGS         0       53    le0
10.0.2.0/24        link#1             UC          0        0    le0
10.0.2.2           52:54:00:12:35:02  UHLW        2       35    le0    796
10.0.2.15          127.0.0.1          UGHS        0        0    lo0
127.0.0.1          127.0.0.1          UH          1        0    lo0
192.168.56.0/24    link#2             UC          0        0    le1
192.168.56.101     08:00:27:00:d4:84  UHLW        1      590    le1   1197
```

The columns shown on these screens indicate various properties of the routes, and are explained in this section.

### Destination

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 by which packets going to a specific destination need to be 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 12.3, “Route Table Flags and Meanings” with some modifications.

### Table 12.3. 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 |
| C | RTF_CLONING | Generate new routes on use |
| c | RTF_PRCLONING | Protocol-specified generate new routes on use |
| D | RTF_DYNAMIC | Created dynamically by redirect |
| G | RTF_GATEWAY | Destination requires forwarding by intermediary |
| H | RTF_HOST | Host entry (net otherwise) |
| L | RTF_LLINFO | Valid protocol to link address translation |
| M | RTF_MODIFIED | Modified dynamically (by redirect) |
| R | RTF_REJECT | Host or net unreachable |
| S | RTF_STATIC | Manually added |
| U | RTF_UP | Route usable |
| W | RTF_WASCLONED | Route was generated as a result of cloning |
| X | RTF_XRESOLVE | External daemon translates proto to link address |

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 flow if this route was used.

## 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 will allow you to view each "hop" along a packet's path as it travels from one end to the other, along with the latency encountered in reaching that intermediate point. On pfSense, you can perform a traceroute by going 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 system's default gateway) will send back the 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 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. It keeps incrementing the TTL 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. ICMP and UDP are layer 4 protocols, and traceroute is done at layer 3, so the protocol used is largely irrelevant except when considering your 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, we will try to find the route to www.google.com:

```
# traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 74.125.95.99
traceroute to www.l.google.com (74.125.95.99), 64 hops max, 40 byte packets
 1  core (172.17.23.1)  1.450 ms  1.901 ms  2.213 ms
 2  172.17.25.21 (172.17.25.21)  4.852 ms  3.698 ms  3.120 ms
 3  bb1-g4-0-2.ipltin.ameritech.net (151.164.42.156)  3.275 ms  3.210 ms  3.215
 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
```

As you can see, it took 10 hops to get there, and the latency generally increases with each hop.

# Routes and VPNs

Depending on the VPN being used, you may or may not see a route showing in the table for the far side. IPsec does not use the routing table, it is instead handled internally in the kernel using the IPsec

SPD. Static routes will never cause traffic to be directed across an IPsec connection. OpenVPN uses the system routing table and as such you will see entries for networks reachable via an OpenVPN tunnel, as in the following example:

```
# netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags  Refs      Use  Netif Expire
default            10.34.29.1         UGS       0 19693837  pppoe0
10.34.29.1         72.69.77.6         UH        1   205590  pppoe0
72.69.77.6         lo0                UHS       0        0     lo0
172.17.212.0/22    192.168.100.1      UGS       0      617  ovpnc1
127.0.0.1          127.0.0.1          UH        0        0     lo0
192.168.10.0/24    link#2             UC        0        0     em0
192.168.100.1      192.168.100.2      UH        3        0  ovpnc1
192.168.130.0/24   192.168.100.1      UGS       0   144143  ovpnc1
192.168.140.0/24   192.168.100.1      UGS       0        0  ovpnc1
```

The OpenVPN interface is 192.168.100.2, with a gateway of 192.168.100.1 and the interface is ovpnc1. There are three networks with OpenVPN pushed routes in that example: 192.168.130.0/24, 192.168.140.0/24, and 172.17.212.0/22.

With IPsec, **traceroute** is not as useful as with routed setups like OpenVPN, because the IPsec tunnel itself does not have IPs. When running **traceroute** to a destination across IPsec, you will see a timeout for the hop that is the IPsec tunnel for this reason.

# Chapter 13. Bridging

Normally each interface on pfSense represents its own broadcast domain with a unique IP subnet, acting the same as separate switches. 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. This is commonly referred to as a transparent firewall.

# Bridging and Layer 2 Loops

When bridging, you need to be careful to avoid layer 2 loops, or have a switch configuration in place that handles them as you desire. A layer 2 loop is when you create the same effect as if you plugged both ends of a patch cable into the same switch. If you have a pfSense install with two interfaces, bridge those interfaces together, then plug both interfaces into the same switch you have created a layer 2 loop. 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 you don't want your network to 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 though, 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.

pfSense 2.x does enable STP on bridge interfaces to help with this, but it can still lead to unexpected situations, like one of your bridge ports shutting itself down to stop the loop.

In a nutshell — bridging has the potential to completely melt down the network you are plugging into if you don't watch what you're plugging in where.

# Creating a Bridge

In pfSense 2.x, bridges are created on Interfaces → (assign) on the Bridge tab. You can add and remove bridges from here. Using bridge entries, you can bind together any number of ports easily. Each bridge you make here will create a new bridge interface in the operating system, named `bridge`*X* 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 will be discussed later in this chapter.

On the Bridge page, click to create a new bridge. You will then be presented with a screen that lets you select the members for the new bridge. Next to Member Interfaces **Ctrl**-click each interface you would like to be part of this bridge, and add a Description if you want. In most cases, you can simply press Save now and be finished.

There are many more advanced options that can be set on a bridge as well. To view these options, click Show Advanced Options. Some of these settings are quite involved, so they will be discussed separately.

## (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. There are quite a few options that control how spanning tree behaves, to 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. You can find more information about STP in the FreeBSD `ifconfig(8)` man page, and on Wikipedia [http://en.wikipedia.org/wiki/Spanning_Tree_Protocol].

### Protocol

This 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.

You may only want to select STP in the case of older switch gear that does not behave well with RSTP.

## STP Interfaces

Here you **Ctrl**-click to select the bridge members which will have STP enabled.

## Valid Time

Set the time that a Spanning Tree Protocol configuration is valid. The default is 20 seconds. The minimum is 6 seconds and the maximum is 40 seconds.

## Forward Time

Set 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.

## Hello Time

Set 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

Set the bridge priority for Spanning Tree. 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

Set the transmit hold count for Spanning Tree. This 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

Sets the Spanning Tree port 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

Sets the Spanning Tree path cost. 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. The default is 100 entries. If there will be a large number of devices communicating across the bridge, you may need to set this higher.

Cache entry expire time controls the timeout of address cache entries in seconds. If set to zero, then address cache entries will not be expired. The default is 240 seconds.

## Span Port

Adds the selected interface as a span port on the bridge. Span ports transmit a copy of every frame received by the bridge. This is most useful for snooping a bridged network passively on another host connected to one of 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 if you are certain 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.

## 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.

# Bridging and Interfaces

In pfSense 2.x, you now have the option to assign the bridge interface (e.g. `bridge0`) itself as another interface. This allows you to assign the bridge as a normal interface and place your IP address on the bridge rather than a member interface, as you had to on pfSense 1.2.x.

In most cases, it is best to put the IP address on the bridge itself. The main reason for this is because IP addresses are dependent on the state of the interface where they are assigned. If you have the IP address for the bridge configured on a member interface and that interface is down, the whole bridge will be down and no longer passing traffic. The most common place that is encountered is a wireless interface bridged to an Ethernet LAN NIC. If the LAN NIC is unplugged, the wireless would be dead, unless the IP assignment was on the bridge interface and not LAN. Another reason is that if you choose to use limiters for controlling traffic, then there must be an IP address on the bridge interface for them to work properly.

## Swapping the 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 you are bridging WLAN to LAN, make the change from WAN or another OPT port. Alternately, you can download a backup of `config.xml` and manually make the changes if you are comfortable doing

that. If you attempt to make changes to a port while managing the firewall from that port, it will most likely result in your access to the GUI being cut off, leaving you unable to reach the firewall.

It is tempting to want to create the bridge and then merely swap the interface assignments, but that won't work because it would end up with the bridge added to itself. For example, if you have LAN and WLAN, create a bridge LANBRIDGE, and then try to swap LAN and LANBRIDGE, it wouldn't work because LAN is specified in the actual bridge setup.

So even though it seems easier to move around the interfaces in this way, it can actually end up being more complicated. In the future, there may be a wizard to address this.

The simplest 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.

If you hand edit your `config.xml` to accomplish this task, you would need to change the LAN assignment to `bridge0`, the former LAN assignment to what used to be the bridge (e.g. OPT2), and then edit the bridge definition to refer to OPT2 and not LAN. You can then restore the configuration, the firewall will reboot, and it should have the desired setup.

## Assigned Bridge MAC Addresses and Windows

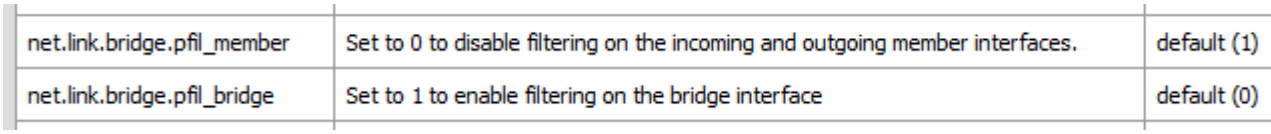
The MAC address for a bridge is determined randomly when the bridge is created, either at boot time or when a new bridge is made. That means that on each reboot, the MAC address can change. In many cases this does not matter, but Windows Vista, 7, and 8 use the MAC address of the gateway to determine if they are on a specific network. If the MAC changes, you will have to re-identify the network as public, private, etc. To work around this, you can enter a MAC address on the assigned bridge interface to spoof it. Then clients will always see the same MAC for the gateway IP.

# Bridging and firewalling

Filtering with bridged interfaces functions similar to routed interfaces, but on pfSense 2.x there are some configuration choices you can make to alter exactly how the filtering behaves. By default, firewall rules are applied on each member interface of the bridge on an inbound basis, just like any other routed interface. Those who have been using pfSense for quite some time will recall an Enable filtering bridge check box on the System → Advanced page. There is outdated information in numerous places referencing this check box. It was inherited from m0n0wall, which did bridging in a different way. Since pfSense uses a different bridging methodology this box is unnecessary, and with the way the bridging methodology in newer FreeBSD versions works it is impossible to have a non-filtering bridge unless you disable pf entirely.

What you can do is 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 13.1, “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*.

**Figure 13.1. Bridge Filtering Tunables**



| | | |
|---|---|---|
| net.link.bridge.pfil_member | Set to 0 to disable filtering on the incoming and outgoing member interfaces. | default (1) |
| net.link.bridge.pfil_bridge | Set to 1 to enable filtering on the bridge interface | default (0) |

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.

# Bridging two internal networks

You can bridge two internal interfaces to combine them on the same broadcast domain and enable filtering on traffic between the two interfaces. This is commonly done with wireless interfaces configured as an access point, to connect the wired and wireless segments on the same broadcast domain. Occasionally a firewall with a LAN and OPT interface will be used in lieu of a switch in networks where only two internal systems are needed. You may encounter scenarios where two interfaces of the firewall need to be on the same broadcast domain for another reason.

### Note

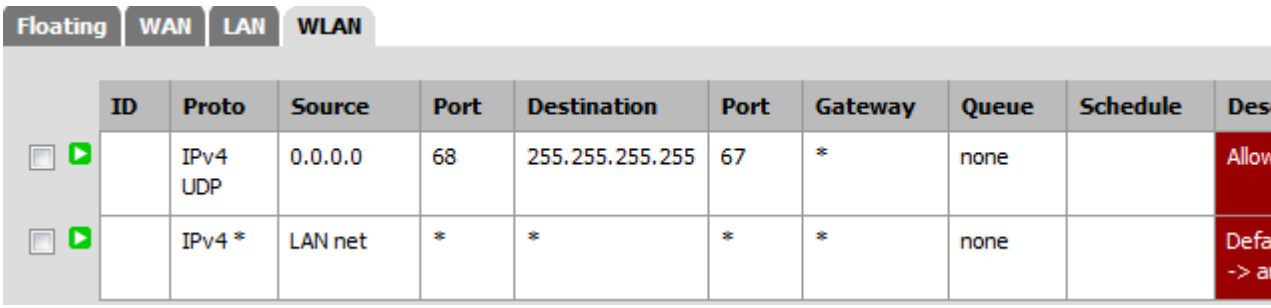
There are additional requirements and restrictions when bridging wireless interfaces because of the way 802.11 functions. See the section called “Bridging and wireless” for more information.

## DHCP and Internal Bridges

If you bridge one internal network to another, two things need to be done. First, ensure that DHCP is only running on the main interface (the one with the IP address) and not the one being bridged. Second, you will need an additional firewall rule at the top of your rules on this OPT interface to allow DHCP traffic.

Normally, when creating a rule to allow traffic on an interface, 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, you must create a rule on the bridged interface with the Protocol set to **`UDP`**, the Source is **`0.0.0.0`**, source port **`68`**, Destination **`255.255.255.255`**, destination port **`67`**. Add a Description stating this will **`Allow DHCP`**, then click Save and Apply Changes. You will end up with a rule that looks like Figure 13.2, “Firewall Rule to Allow DHCP”.

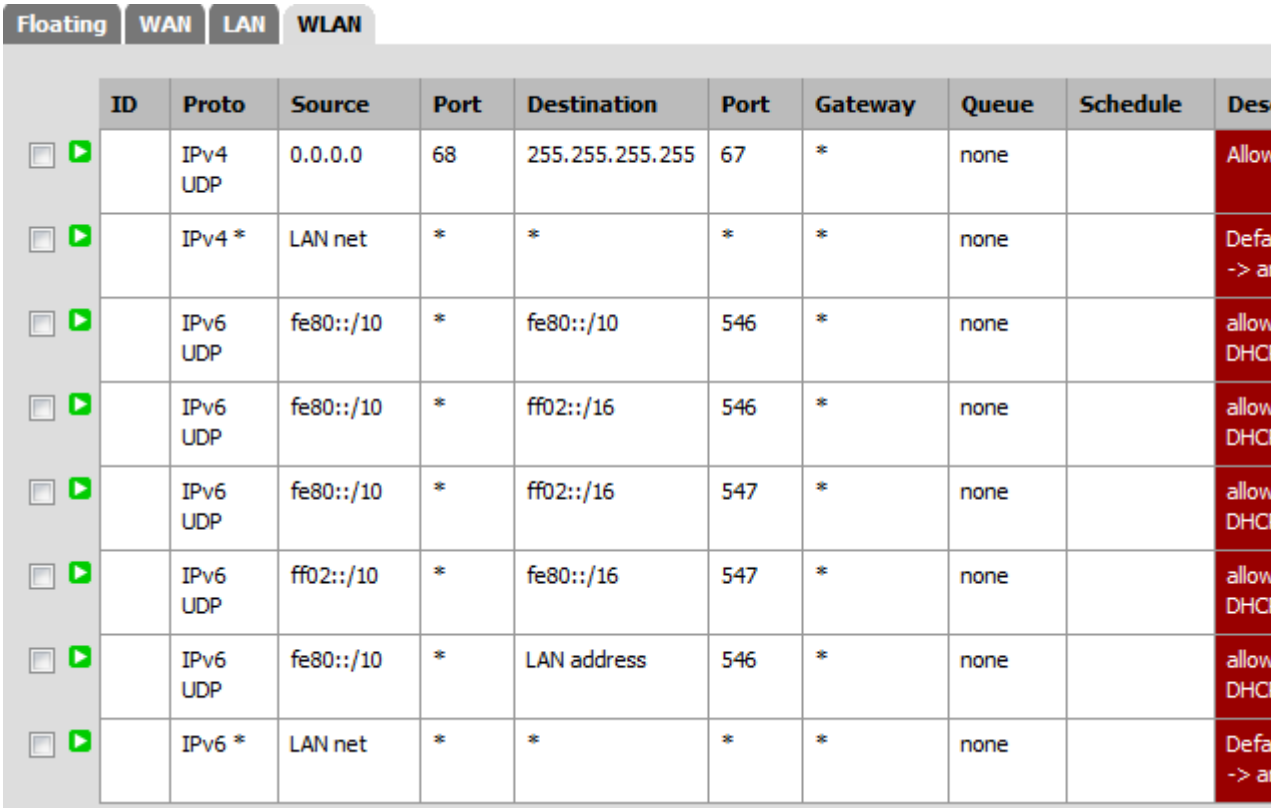
**Figure 13.2. Firewall Rule to Allow DHCP**




After adding that rule, clients in the bridged segment should 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 13.3, “Firewall Rule to Allow both DHCP and DHCPv6” for the list of rules you might need. These can be simplified with aliases into one or two rules containing the proper source network, destination network, and ports.

**Figure 13.3. Firewall Rule to Allow both DHCP and DHCPv6**




| | ID | Proto | Source | Port | Destination | Port | Gateway | Queue | Schedule | Des |
|---|---|---|---|---|---|---|---|---|---|---|
| | | IPv4 UDP | 0.0.0.0 | 68 | 255.255.255.255 | 67 | * | none | | Allow |
| | | IPv4 * | LAN net | * | * | * | * | none | | Defa -> a |
| | | IPv6 UDP | fe80::/10 | * | fe80::/10 | 546 | * | none | | allow DHC |
| | | IPv6 UDP | fe80::/10 | * | ff02::/16 | 546 | * | none | | allow DHC |
| | | IPv6 UDP | fe80::/10 | * | ff02::/16 | 547 | * | none | | allow DHC |
| | | IPv6 UDP | ff02::/10 | * | fe80::/16 | 547 | * | none | | allow DHC |
| | | IPv6 UDP | fe80::/10 | * | LAN address | 546 | * | none | | allow DHC |
| | | IPv6 * | LAN net | * | * | * | * | none | | Defa -> a |

# Bridging OPT to WAN (Transparent Bridge)

Bridging an OPT interface with WAN, also known as a "transparent bridge" allows you to use public IPs on your internal network that have a gateway IP residing on your WAN network. One situation where this is common is for DHCP assigned public IP addresses. You can use pfSense to protect systems that obtain public IPs directly from your ISP's DHCP server by using a bridged interface. This is also useful in scenarios with a single public IP block where you need public IPs directly assigned to hosts, as described in the section called “Single IP subnet”.

In a transparent bridge scenario you must ensure the clients behind the bridge still use the WAN gateway on the outside of the bridge, and *not* any IP address that pfSense may have on the bridge.

# Bridging interoperability

Since bridged interfaces behave differently than normal interfaces in some regards, there are a few things 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 (Chapter 24, *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 you can't, for example, bridge LAN to WAN and hope to capture clients with the portal.

In pfSense 2.0 and later this can work if you are bridging multiple local interfaces to all route through pfSense (e.g. LAN1, LAN2, LAN3, etc). If you assign the bridge interface, give it an IP, and that IP is used as the gateway by clients on the bridge, then it can function as expected.

# High Availability

High availability (Chapter 25, *Firewall Redundancy / High Availability*) is not recommended with bridging at this time — but, there are some manual hacks. Using HA with networks that involve bridging is not generally recommended, but this kind of setup has worked for a number of individuals. 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. If STP is not available, there are two other approaches for handling a bridge in this scenario, similar but not as elegant as STP. Both of these methods require changing files on the pfSense system, and would not survive a backup/restore without special consideration. These techniques are a **cron** script to manage the bridge, or a **devd** hook to manage the bridge. Both of these methods are described in a sticky post on the CARP/VIP forum [http://forum.pfsense.org/index.php/topic,4984.0.html] and later in this chapter.[1]

## Configure your primary and backup firewalls

Configure your primary and backup firewalls as you would with any HA deployment, as covered in Chapter 25, *Firewall Redundancy / High Availability*. Configure the bridge interface on both the primary and secondary, using the same interface description. If the bridge is OPT1 on the primary, make it OPT1 on the secondary. Do not plug in both bridges simultaneously until the end. You will need to be able to access the pfSense WebGUI from a firewall interface other than the bridge interface. You will need to perform all of these steps for both your primary and secondary firewalls.

## Configuring STP

Even with STP active, some configuration will be needed on the switch in order to nudge STP into making the right choice about which port should be kept open and which should be blocked. Otherwise you could end up with a situation where the traffic is actually flowing through your backup router's bridge instead of the primary router, leading to unpredictable behavior. Port blocking in this situation is controlled by setting port priorities and path costs.

On a Cisco switch, the configuration would look something like this:

```
interface FastEthernet0/1
  description Firewall - Primary - DMZ Port
  switchport access vlan 20
  spanning-tree vlan 20 port-priority 64
  no cdp enable

interface FastEthernet0/2
  description Firewall - Backup - DMZ Port
  switchport access vlan 20
  spanning-tree vlan 20 cost 500
  no cdp enable
```

By giving the primary's port a lower than normal priority (64 vs. the default 128), it will be more likely to be used, especially given the higher path cost (500 vs. the default 19) of the other port. These values can be checked as follows (on the switch):

```
# show spanning-tree interface FastEthernet0/1
Interface FastEthernet0/1 (port 13) in Spanning tree 20 is FORWARDING
```

[1] http://forum.pfsense.org/index.php/topic,4984.0.html

```
Port path cost 19, Port priority 64
Designated root has priority 32768, address 0002.4b6e.xxxx
Designated bridge has priority 32768, address 0002.b324.xxxx
Designated port is 3, path cost 131
Timers: message age 6, forward delay 0, hold 0
BPDU: sent 18411032, received 16199798

# show spanning-tree interface FastEthernet0/2
Interface FastEthernet0/2 (port 14) in Spanning tree 20 is BLOCKING
Port path cost 500, Port priority 128
Designated root has priority 32768, address 0002.4b6e.xxxx
Designated bridge has priority 32768, address 0002.b324.xxxx
Designated port is 4, path cost 131
Timers: message age 6, forward delay 0, hold 0
BPDU: sent 434174, received 15750118
```

As you can see, the primary system's switch port is forwarding is it should be, and the backup port is blocking. If traffic stops flowing through the primary port, the backup should switch to a forwarding state.

Switches from other vendors support similar functionality. Refer to your switch's documentation for information on STP configuration.

In pfSense 2.0, STP can be configured and handled directly on a bridged interface by editing the entry for a specific bridge under Interfaces → (assign) on the Bridge tab.

# High Availability check script for cron

In this method, a script runs from **cron** every minute and checks to see if the system is MASTER or BACKUP in the HA cluster. If the system is MASTER, the bridge is brought up, if the system is BACKUP, the bridge is put into a down state. It prevents the loop by only having one bridge active at any given time, but as you can probably tell by how often the **cron** script runs, there may be as much as a minute of downtime for bridged systems before the script detects the switch and activates the backup bridge.

## Add the Script

First you need to add a script to check your CARP status and modify your bridge status accordingly. The following provides an example that can be used. It is also available for download [http://files.pfsense.org/misc/bridgecheck.sh].

```
#!/bin/sh
#
# CARP check script for bridging
#
if ifconfig wan_vip1 | grep BACKUP > /dev/null 2>&1 ; then
        /sbin/ifconfig bridge0 down
else
        /sbin/ifconfig bridge0 up
fi
```

Copy that script somewhere, for example, `/usr/bin/bridgecheck.sh`. The following command will download this file from files.pfsense.org and save it as `/usr/bin/bridgecheck.sh`.

```
# fetch -o /usr/bin/bridgecheck.sh \
     http://files.pfsense.org/misc/bridgecheck.sh
```

Then you need to make the script executable by running the following command.

```
# chmod +x /usr/bin/bridgecheck.sh
```

After that, be sure to edit the script and adjust the check to match a VIP that exists on your firewall.

### Schedule the script

Now you need to schedule the script to run. Install the cron package from System → Packages and then you can add a cron job under Services → Cron that looks like .

**Figure 13.4. Add Interface Group**


Make sure to change both the primary and the secondary.

### Disable bridge at boot

You will want to add a command to the configuration to down the bridge at boot time. This will help prevent layer 2 loops, as **bridgecheck.sh** will bring the HA master's bridge online within 1 minute. Install the Shellcmd package from System → Packages and then from Services → Shellcmd you can add a shellcmd command entry for **`/sbin/ifconfig bridge0 down`**.

As with the cron entry before, make the change on both the primary and the secondary.

## devd Hooks

On pfSense 2.0 and later, there are predefined **devd** entries that catch the actual CARP state transition as it happens. These transitions are setup to trigger a script depending on the new state of the VIP. These scripts are **/etc/rc.carpmaster** and **/etc/rc.carpbackup**. You can edit these scripts and have them handle the bridge state for you.

Edit each of those files, adding a line at the end, like this for : `/etc/rc.carpmaster`

```
shell_exec("/sbin/ifconfig bridge0 up");
```

And: `/etc/rc.carpbackup`

```
shell_exec("/sbin/ifconfig bridge0 down");
```

That will automatically bring the bridge up and down any time a CARP state change is detected.

## Troubleshooting failover bridging

If something is not working as intended, check the Status → Interfaces page on both systems to review the `bridge0` interface, and the CARP status page to verify CARP's master or backup status. You can run **bridgecheck.sh** from the command line, as well as checking interface status using **ifconfig**. An understanding of the underlying FreeBSD OS may be necessary to successfully troubleshoot any problems with this type of deployment.

Many problems with HA and Bridging will arise from switch loops and STP issues. Go over the section called “High Availability” again, and also check your switch configuration to see the port status for your bridged interfaces. If your ports are blocking when they should be forwarding, you will probably need to adjust STP settings or employ one of the alternate techniques to shut down a backup bridge.

# Multi-WAN

Bridging by its nature is incompatible with multi-WAN in many of its uses. When using bridging, commonly something other than pfSense will be the default gateway for the hosts on the bridged interface, and that router is the only thing that can direct traffic from those hosts. This doesn't prevent you from using multi-WAN 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 you bridge multiple internal interfaces together and pfSense is the default gateway for your hosts on a bridged interface, then you can use multi-WAN the same as with non-bridged interfaces.

# Chapter 14. Virtual LANs (VLANs)

VLANs provide a means of segmenting a single switch into multiple broadcast domains, allowing a single switch to function the same as multiple switches. This is commonly used for network segmentation in the same way that multiple switches could be used, to place hosts on a specific segment as configured on the switch. Where trunking is employed between switches, devices on the same segment need not reside on the same switch. The concepts, terminology and configuration of VLANs are all covered in this chapter.

# 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 since about the year 2000 supports 802.1Q VLAN trunking. You cannot use VLANs with an unmanaged switch.

2. Network adapter capable of VLAN tagging — you will need a NIC that supports hardware VLAN tagging or has long frame support. Because each frame has a 4 byte 802.1Q tag added in the header, 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

Just because an adapter is listed as having long frame support does not guarantee your NIC's specific implementation of that 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 you encounter problems using one of the NICs listed under long frame support, trying an interface with VLAN hardware tagging support is recommended. We are not aware of any similar problems with NICs listed under VLAN hardware support.

**Ethernet interfaces with VLAN hardware support**:

`bce(4)`, `bge(4)`, `cxgb(4)`, `em(4)`, `ixgb(4)`, `msk(4)`, `nge(4)`, `re(4)`, `stge(4)`, `ti(4)`, `txp(4)`, `vge(4)`.

**Ethernet interfaces with long frame support**:

`bfe(4)`, `dc(4)`, `fxp(4)`, `gem(4)`, `hme(4)`, `le(4)`, `nfe(4)`, `nve(4)`, `rl(4)`, `sis(4)`, `sk(4)`, `ste(4)`, `tl(4)`, `tx(4)`, `vr(4)`, `xl(4)`

# Terminology

This section covers the terminology you will need to understand to successfully deploy VLANs.

## Trunking

Trunking refers to a means of carrying multiple VLANs on the same 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 only the router providing connectivity between the VLANs, in this case, pfSense, as well as any connections to other switches containing multiple VLANs.

## VLAN ID

Each VLAN has an ID associated with it that is used for identification of 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 the section called “VLANs and Security”. Aside from avoiding the use of VLAN 1, you can choose which VLAN numbers you wish to use. Some will start with VLAN 2 and increment by one until the required number of VLANs is reached. Another common practice is using the third octet in the IP subnet of the VLAN as the VLAN ID. For example, if you use 10.0.10.0/24, 10.0.20.0/24 and 10.0.30.0/24, it is logical to use VLANs 10, 20, and 30 respectively. Choose a VLAN ID assignment scheme that makes sense to you.

## Parent interface

The parent interface refers to the physical interface where the VLANs reside, such as `em0` or `bge0`. When you configure VLANs on pfSense or FreeBSD, each is assigned a virtual interface, starting with `vlan0` and incrementing by one for each additional VLAN configured. In pfSense 1.2.x, the number of the VLAN interface has no correlation to the VLAN ID. You *should not* assign your parent interface to any interface on pfSense — its sole function should be as the parent for the defined VLANs. In some situations this will work, but can cause difficulties with switch configuration, can cause problems with using Captive Portal, and forces you to use the trunk port's default VLAN, which should be avoided as discussed further in the section called “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. You connect every device residing on a single VLAN to an access port. Most of your switch ports will be configured as access ports. Devices on access ports are unaware of any VLANs in your network. They see each VLAN the same as they would a switch without VLANs.

## Double tagging (QinQ)

It is also possible to double tag traffic, using both an outer and inner 802.1Q tag. This is referred to as *QinQ*. This can be useful in large ISP environments and some other very large networks. Triple tagging is also possible. pfSense does support QinQ as of pfSense 2.0. 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 the section called “pfSense QinQ Configuration”.

## Private VLAN (PVLAN)

PVLAN 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 users have. 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 offer a great means to segment your network and isolate subnetworks, but there are some 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 your network vulnerable. There have also been past security problems in switch vendors' implementations of VLANs.

# Segregating Trust Zones

Because of the possibility of misconfiguration, you should segregate networks of considerably different trust levels onto their own physical switches. For example, while you could technically use the same switch with VLANs for all your internal networks as well as the network outside your firewalls, that should be avoided as a simple misconfiguration of the switch could lead to unfiltered Internet traffic entering your internal network. At a minimum, you should 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 your network depends on your specific circumstances, and level of risk and paranoia.

# Using the default VLAN1

Because VLAN1 is the default, or "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 you need. It is always better to use a different VLAN, and ensure that you only select the ports you want on your switch group to 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 your switches use these protocols. It is generally best to keep that internal traffic isolated from your data traffic.

If you must use VLAN1, you must take great care to assign every single port on every switch to a different VLAN except those you want in VLAN1, and do not create a management interface for the switch on VLAN1. You should also change the native VLAN of the switch group to a different, unused, VLAN. Some switches may not support any of these workarounds, and so it is typically easier to move your data to a different VLAN instead of fussing with making VLAN1 available. With VLAN ID 2 through 4094 to choose from, it is undoubtedly better to just ignore VLAN1 when designing your 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 your switch configurations, it is important to physically secure trunk ports. Also make sure there are no ports configured for trunking that are left unplugged where someone could hook into one, accidentally or otherwise. Depending on your switch, it may support dynamic negotiation of trunking. You should ensure this functionality is disabled or properly restricted.

# Other Issues with Switches

There have been reports that some VLAN based switches will leak traffic across VLANs when they come 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 you have, do some research online to see if it has undergone any kind of security testing, and ensure you are using the latest firmware. While these issues are a problem with the switch, and not pfSense, they are part of your overall security.

Many of the things here are specific to particular makes and models of switches. There may be different security considerations specific to the switch you are using. Refer to its documentation for recommendations on VLAN security.

# pfSense VLAN Configuration

This section covers the configuration of VLANs on the pfSense side.

## Console VLAN configuration

You can configure VLANs at the console using the Assign Interfaces function. The following example shows how to configure two VLANs, ID 10 and 20, with `le2` as the parent interface. The VLAN interfaces are assigned as OPT1 and OPT2.

```
 pfSense console setup
**********************
 0) Logout (SSH only)                  8) Shell
 1) Assign Interfaces                  9) pfTop
 2) Set interface(s) IP address       10) Filter Logs
 3) Reset webConfigurator password    11) Restart webConfigurator
 4) Reset to factory defaults         12) pfSense Developer Shell
 5) Reboot system                     13) Upgrade from console
 6) Halt system                       14) Disable Secure Shell (sshd)
 7) Ping host                         15) Restore recent configuration


Enter an option: 1


Valid interfaces are:

vr0   00:0d:b9:01:aa:a0   (up) VIA VT6105M Rhine III 10/100BaseTX
vr1   00:0d:b9:01:aa:a1   (up) VIA VT6105M Rhine III 10/100BaseTX
vr2   00:0d:b9:01:aa:a2 (down) VIA VT6105M Rhine III 10/100BaseTX
ath0  90:a4:de:02:ff:bb   (up) Atheros 9280

Do you want to set up VLANs first?

If you are not going to use VLANs, or only for optional interfaces, you should
say no here and use the webConfigurator to configure VLANs later, if required.

Do you want to set up VLANs now [y|n]? y

VLAN Capable interfaces:

vr0     00:0d:b9:18:8a:a0   (up)
vr1     00:0d:b9:18:8a:a1   (up)
vr2     00:0d:b9:18:8a:a2

Enter the parent interface name for the new VLAN (or nothing if finished): vr2
Enter the VLAN tag (1-4094): 10

VLAN Capable interfaces:
```

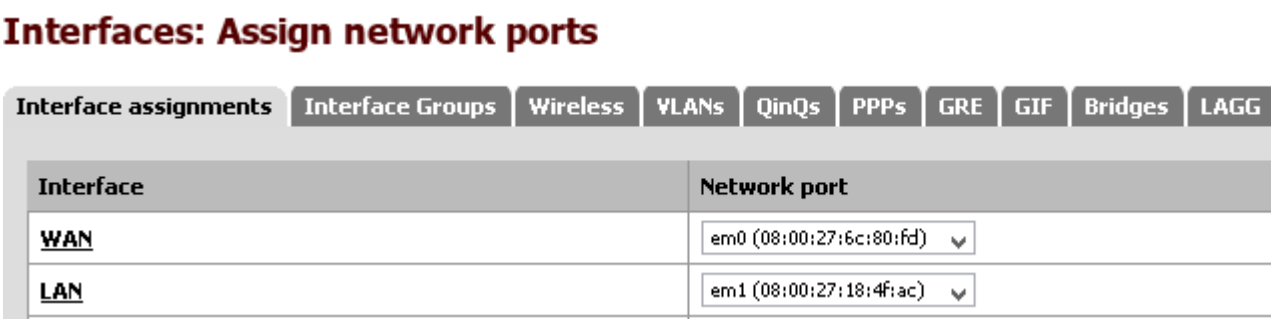
```
vr0     00:0d:b9:18:8a:a0   (up)
vr1     00:0d:b9:18:8a:a1   (up)
vr2     00:0d:b9:18:8a:a2

Enter the parent interface name for the new VLAN (or nothing if finished): vr2
Enter the VLAN tag (1-4094): 20

VLAN Capable interfaces:

vr0     00:0d:b9:18:8a:a0   (up)
vr1     00:0d:b9:18:8a:a1   (up)
vr2     00:0d:b9:18:8a:a2

Enter the parent interface name for the new VLAN (or nothing if finished): <ente


VLAN interfaces:

vr2_vlan10      VLAN tag 10, parent interface vr2
vr2_vlan20      VLAN tag 20, parent interface vr2

*NOTE*  pfSense requires *AT LEAST* 1 assigned interface(s) to function.
        If you do not have *AT LEAST* 1 interfaces you CANNOT continue.

        If you do not have at least 1 *REAL* network interface card(s)
        or one interface with multiple VLANs then pfSense
        *WILL NOT* function correctly.

If you do not know the names of your interfaces, you may choose to use
auto-detection. In that case, disconnect all interfaces now before
hitting 'a' to initiate auto detection.

Enter the WAN interface name or 'a' for auto-detection: vr1

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(or nothing if finished): vr0

Enter the Optional 1 interface name or 'a' for auto-detection
(or nothing if finished): vr2_vlan10

Enter the Optional 2 interface name or 'a' for auto-detection
(or nothing if finished): vr2_vlan20

Enter the Optional 3 interface name or 'a' for auto-detection
(or nothing if finished): <enter>

The interfaces will be assigned as follows:

LAN  -> vr1
WAN  -> vr0
OPT1 -> vr2_vlan10
OPT2 -> vr2_vlan20

Do you want to proceed [y|n]? y

One moment while we reload the settings...
```

After a few seconds, your settings will reload and you will be returned to the console menu. When configuring VLAN interfaces at the console, it doesn't warn you about the reboot that may be needed before VLANs will function. Some rare network adapters or drivers will not work properly with VLANs until the system is rebooted. This isn't normally necessary, especially on pfSense 2.x. If you are still running pfSense 1.x, then to be on the safe side, rebooting after your initial VLAN setup is recommended. For future VLAN additions once VLANs are already configured, a reboot is not required.

# Web interface VLAN configuration

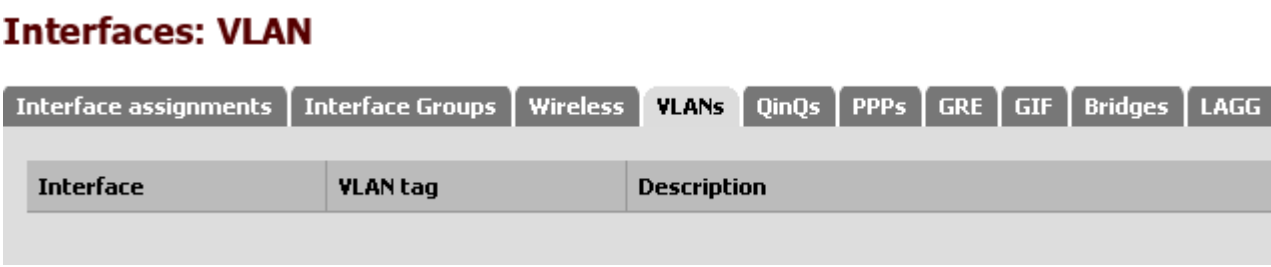
Browse to Interfaces → (assign). Figure 14.1, “Interfaces: Assign” shows the system being used for this example. WAN and LAN are assigned as `em0` and `em1` respectively. There is also a `em2` interface that will be used as the VLAN parent interface.

**Figure 14.1. Interfaces: Assign**



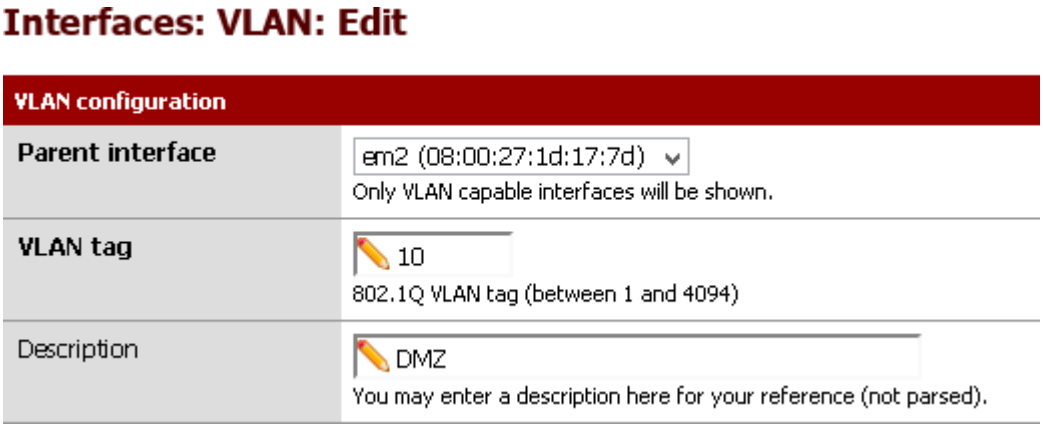

Click the VLANs tab. Then click to add a new VLAN, as shown in Figure 14.2, “VLAN List”.

**Figure 14.2. VLAN List**



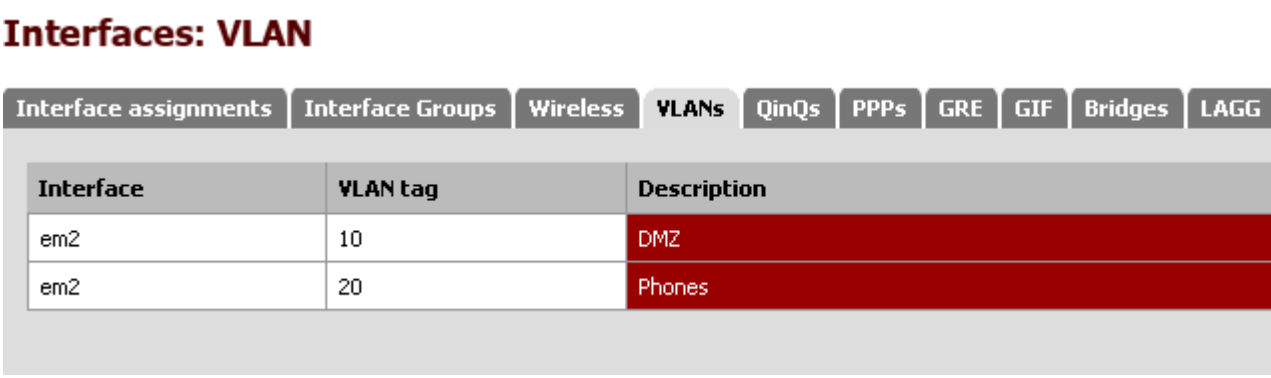

The VLAN editing screen should now be shown, like Figure 14.3, “Edit VLAN”. From here, pick a Parent Interface, **`le2`**. Then enter a VLAN tag, **`10`**, and enter a Description, such as what network is on that VLAN (DMZ, Databases, testing, etc.).

**Figure 14.3. Edit VLAN**



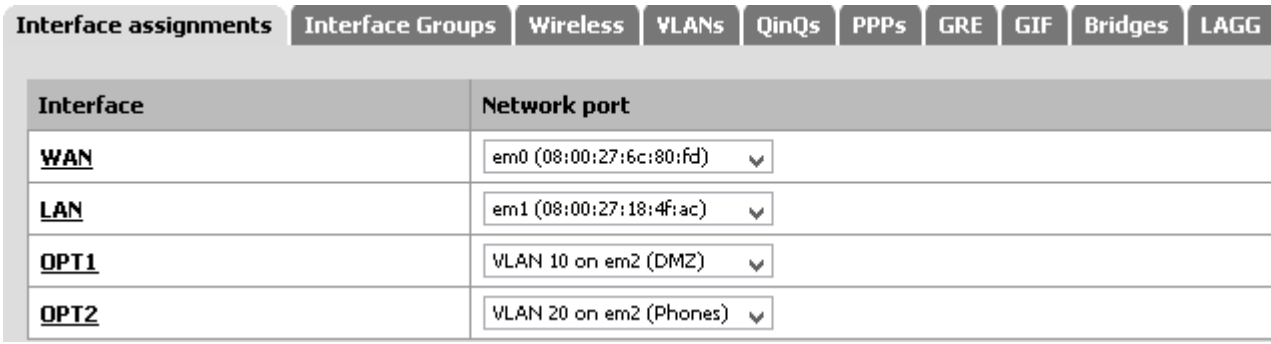
Once Save is clicked, you will return to the list of available VLANs, which should now include the newly added VLAN 10. Repeat that process to add additional VLANs, such as VLAN 20. These can be seen in Figure 14.4, “VLAN List”

**Figure 14.4. VLAN List**




Now, to assign the VLANs to interfaces, click the Interface Assignments tab, then click , and in the drop-down list of available interface assignments, you should see the new VLANs. For OPT1, pick the interface with VLAN ID 10. Click again, and for OPT2, pick the interface with VLAN ID 20. When finished, it will look something like Figure 14.5, “Interface list with VLANs”

**Figure 14.5. Interface list with VLANs**




The VLAN-based OPT interfaces behave as any other OPT interfaces do, which means they must be enabled, configured, firewall rules added, and services like the DHCP Server will need to be configured if needed. See the section called “Interface Configuration Basics” for more information on configuring optional interfaces.

# Switch VLAN Configuration

This section provides guidance on configuring your switch. 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 you will need for VLANs to function, and it does not necessarily show the ideal secure switch configuration for your environment. An in depth discussion of switch security is outside the scope of this book.

## Switch configuration overview

Generally you will need to configure three or four things on VLAN capable switches.

1. Add/define the VLANs — most switches have a means of adding VLANs, and they must be added before they can be configured on any ports.

2. Configure the trunk port — configure the port pfSense will be connected to as a trunk port, tagging all your VLANs on the interface.

3. Configure the access ports — configure the ports your internal hosts will be using as access ports on the desired VLANs, with untagged VLANs.

4. Configure the Port VLAN ID (PVID) — some switches require configuring the PVID for a port. 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 you to configure this in two places. Check your switch's 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 your VLAN configuration. With VTP, if you decide you need another VLAN it only needs added to a single switch, and then all other trunked switches in the group can assign ports to that VLAN. If VLANs are configured independently, you must add them to each switch by hand. Refer to Cisco's documentation on VTP to ensure you have a secure configuration not prone to accidental destruction. In a network with only a few switches where VLANs do not change frequently, you are usually better off not using VTP to 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 your 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 FastEthernet0/24
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 your switch does not allow the **encapsulation dot1q** configuration option, it only supports 802.1Q and you need not worry about specifying the encapsulation.

## Add Ports to the VLAN

To add ports to these VLANs, you need to assign them as follows:

```
sw# configure terminal
sw(config)# interface FastEthernet0/12
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. You still have the option of using standalone VLANs or VTP or 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
```

And 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 consideration is needed there. First, 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 you envision needing.

6. Restart the switch to apply the changes.

## Create VLANs

Before the VLANs can be assigned to ports, you need to create the VLANs. 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, ***LAN***

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 you want to assign

6. Press **space** on Default VLAN until it says No

7. Move over to the column for each of the VLANs on this trunk port, and Press **space** until it says 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 you want to assign
6. Press **space** on Default VLAN until it says No
7. Move over to the column for the VLAN to which this port will be assigned
8. Press **space** until it says Untagged.

# Netgear managed switches

This example is on a GS108T, but other Netgear models we have seen 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 your switch's web interface to start.

## Planning the VLAN configuration

Before configuring the switch, you need to know how many VLANs you will configure, what IDs you will use, and how each switch port needs to be configured. For this example, we are using an 8 port GS108T, and will be configuring it as shown in Table 14.1, “Netgear GS108T VLAN Configuration”.

### Table 14.1. 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 |

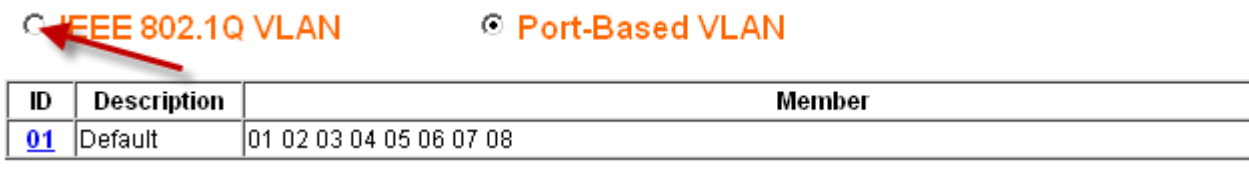
## Enable 802.1Q VLANs

In the System menu on the left side of the page, click VLAN Group Setting, as indicated in Figure 14.6, “VLAN Group Setting”.

### Figure 14.6. VLAN Group Setting

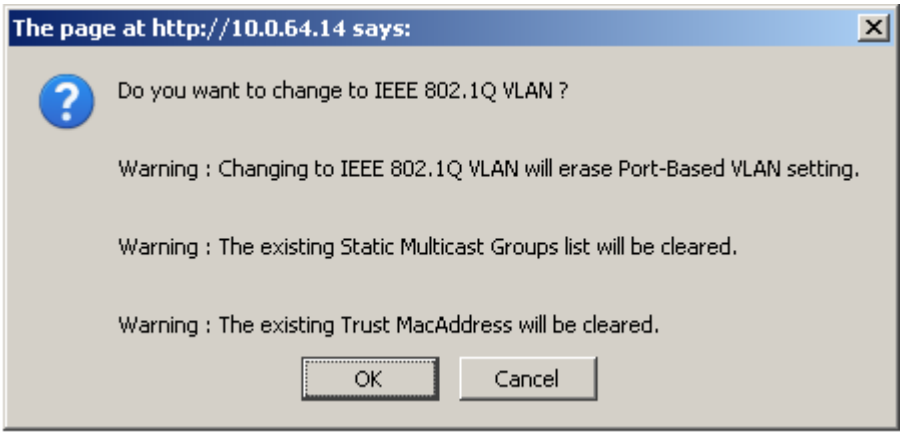

Select IEEE 802.1Q VLAN (Figure 14.7, “Enable 802.1Q VLANs”).

### Figure 14.7. Enable 802.1Q VLANs



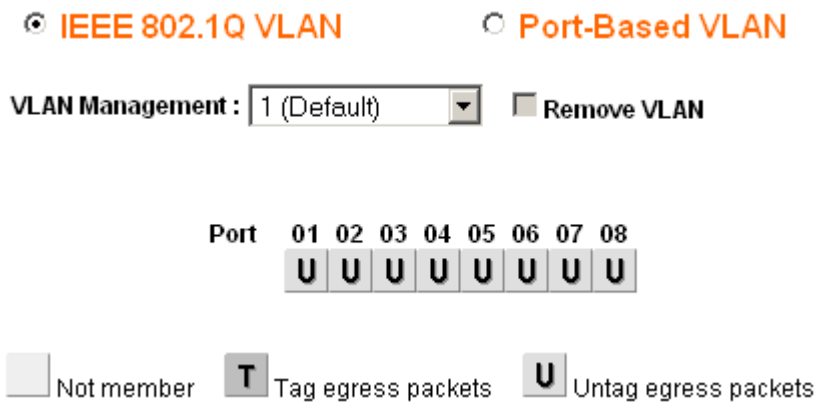
This will prompt you with a pop up asking if you really want to change, and listing some of the consequences, as shown in Figure 14.8, “Confirm change to 802.1Q VLAN”. If you want to trunk VLANs, you must use 802.1Q. Click OK.

### Figure 14.8. Confirm change to 802.1Q VLAN




After clicking OK, the page will refresh with your 802.1Q VLAN configuration as shown in Figure 14.9, “Default 802.1Q configuration”.

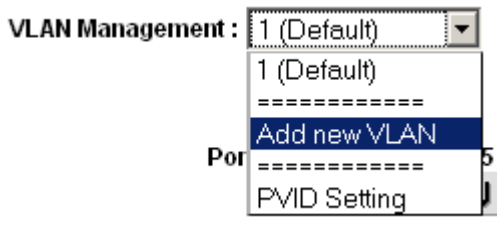
### Figure 14.9. Default 802.1Q configuration




## Add VLANs

For this example, I will add two VLANs with IDs *10* and *20*. To add a VLAN, click the VLAN Management drop down and click Add new VLAN as shown in Figure 14.10, “Add new VLAN”.

### Figure 14.10. Add new VLAN




Enter the VLAN ID for this new VLAN, then click Apply. The VLAN screen will now let you configure VLAN *10* (Figure 14.11, “Add VLAN 10”). Before configuring it, I will again click Add

new VLAN as shown in Figure 14.10, “Add new VLAN” to add VLAN *20* (Figure 14.12, “Add VLAN 20”).

**Figure 14.11. Add VLAN 10**

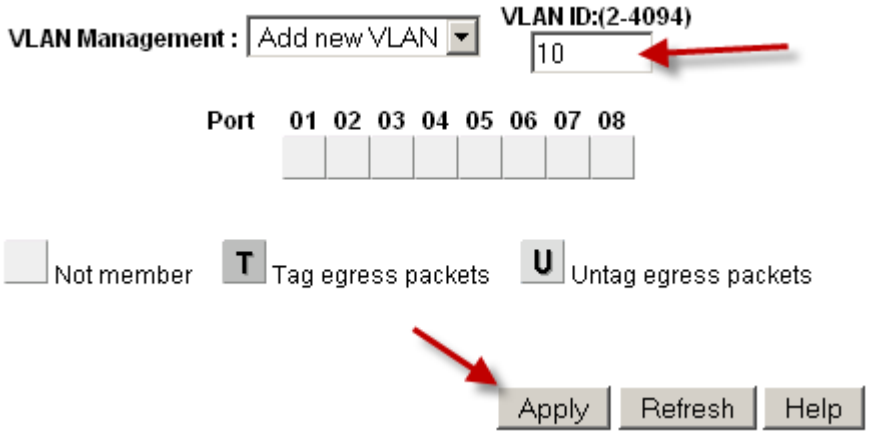
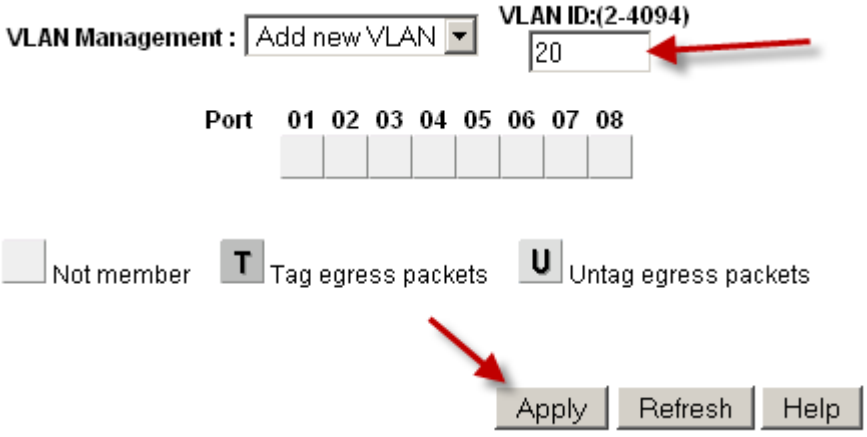



**Figure 14.12. Add VLAN 20**




Add as many VLANs as you need, then continue to the next section.

## Configure VLAN tagging

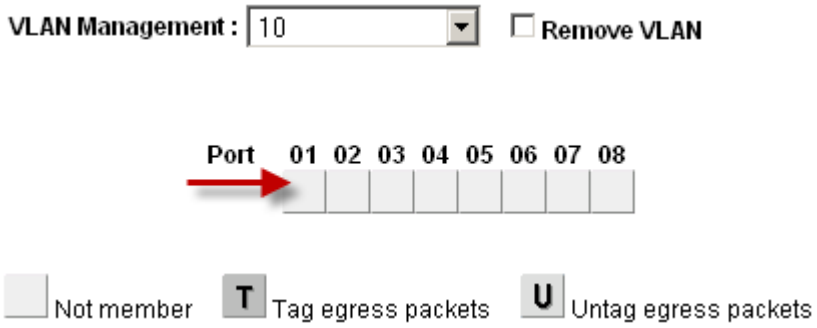
When you select a VLAN from the VLAN Management drop down, it shows you how that VLAN is configured on each port. A blank box means the port is not a member of the selected VLAN. A box containing **T** means the VLAN is sent on that port with the 802.1Q tag. **U** indicates the port is a member of that VLAN and it leaves the port untagged. The trunk port will need to have both VLANs added and tagged.

### Note

Do not change the configuration of the port you are using to access the switch's web interface. You would lock yourself out, with the only means of recovery on the GS108T is hitting the reset to factory defaults button — it doesn't have a serial console. For the switches that have serial consoles, have a null modem cable handy in case you disconnect yourself from network connectivity with the switch. Configuring the management VLAN is covered later in this section.

Click in the boxes beneath the port number as shown in Figure 14.13, “Toggle VLAN membership” to toggle between the three VLAN options.

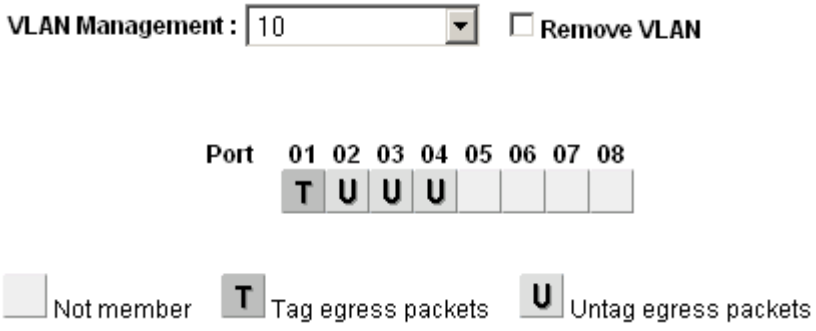
**Figure 14.13. Toggle VLAN membership**




## Configure VLAN 10 membership

Figure 14.14, “Configure VLAN 10 membership” shows VLAN *10* configured as outlined in Table 14.1, “Netgear GS108T VLAN Configuration”. The access ports on this VLAN are set to untagged while the trunk port is set to tagged.

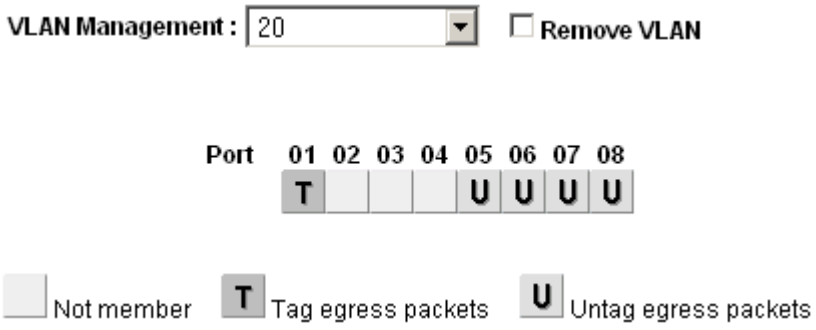
**Figure 14.14. Configure VLAN 10 membership**




## Configure VLAN 20 membership

Select *20* from the VLAN Management drop down to configure the port memberships for VLAN *20*.

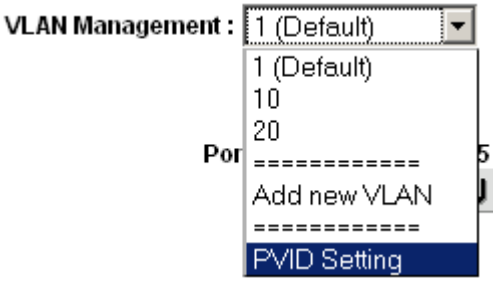
**Figure 14.15. Configure VLAN 20 membership**




## Change PVID

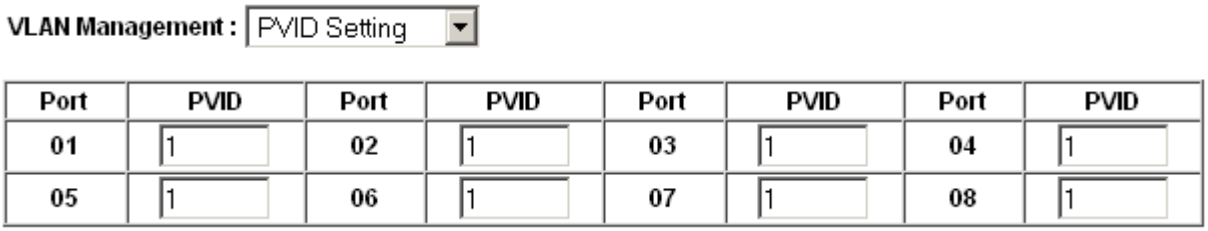
On Netgear switches, in addition to the previously configured tagging settings, you must also configure the PVID to specify the VLAN used for frames entering that port. In the VLAN Management drop down, click PVID Setting as shown in Figure 14.16, “PVID Setting”.

### Figure 14.16. PVID Setting



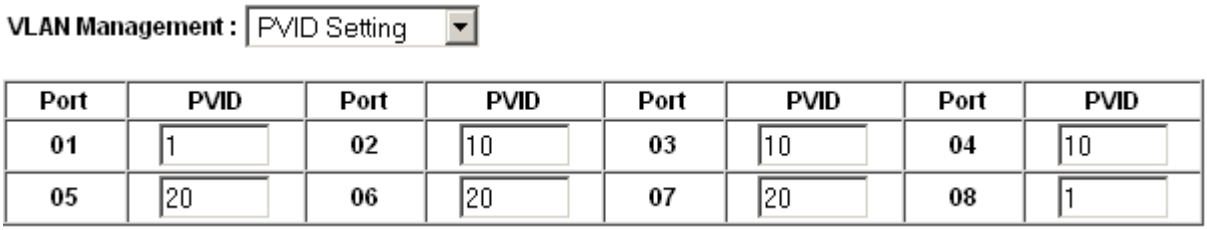

The default PVID setting is VLAN 1 for all ports as shown in Figure 14.17, “Default PVID Configuration”.

### Figure 14.17. Default PVID Configuration




Change the PVID for each access port, but leave the trunk port and port you are using to access the switch's management interface set to **1**. Figure 14.18, “VLAN 10 and 20 PVID Configuration” shows the PVID configuration matching the port assignments shown in Table 14.1, “Netgear GS108T VLAN Configuration”, with port 8 being used to access the switch's management interface. Apply your changes when finished.

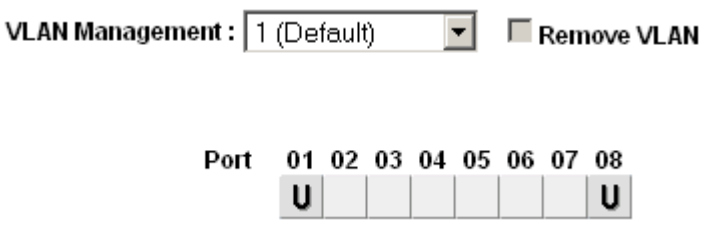
### Figure 14.18. VLAN 10 and 20 PVID Configuration




## Remove VLAN 1 configuration

By default, all ports are members of VLAN 1 with untagged egress frames. Select **1 (Default)** from the VLAN Management drop down. Remove VLAN 1 from all ports *except* the one you are using to manage the switch and the trunk port, so you don't get disconnected. I am using port 8 to manage the switch. When finished, your screen should look like Figure 14.19, “Remove VLAN 1 membership”.

### Figure 14.19. Remove VLAN 1 membership



Apply your changes when finished.

### Verify VLAN functionality

Configure your VLANs on pfSense, including the DHCP server on the VLAN interfaces if you will be using DHCP. 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 your tagging and PVID configuration on the switch, and your 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 you might find on an ISP, Metro Ethernet, or datacenter link between locations. It can be a quick/easy way of trunking VLANs across locations without having a trunking-capable connection between the sites.

Setting up QinQ interfaces on pfSense is fairly simple. First, go to Interfaces → (assign) on the QinQ tab, and click .

On this screen, select the Parent Interface that will carry the QinQ traffic.

Next, enter the First level tag. This is the outer VLAN ID on the QinQ interface, or the VLAN ID given by your provider for the site-to-site link.

If you check Adds interface to QinQ interface groups so you can write filter rules easily, then a new interface group will be created called QinQ that can be used to filter all of the QinQ subinterfaces at once. When you have hundreds or potentially thousands of QinQ tags, this greatly reduces the amount of work needed to use the QinQ interfaces.

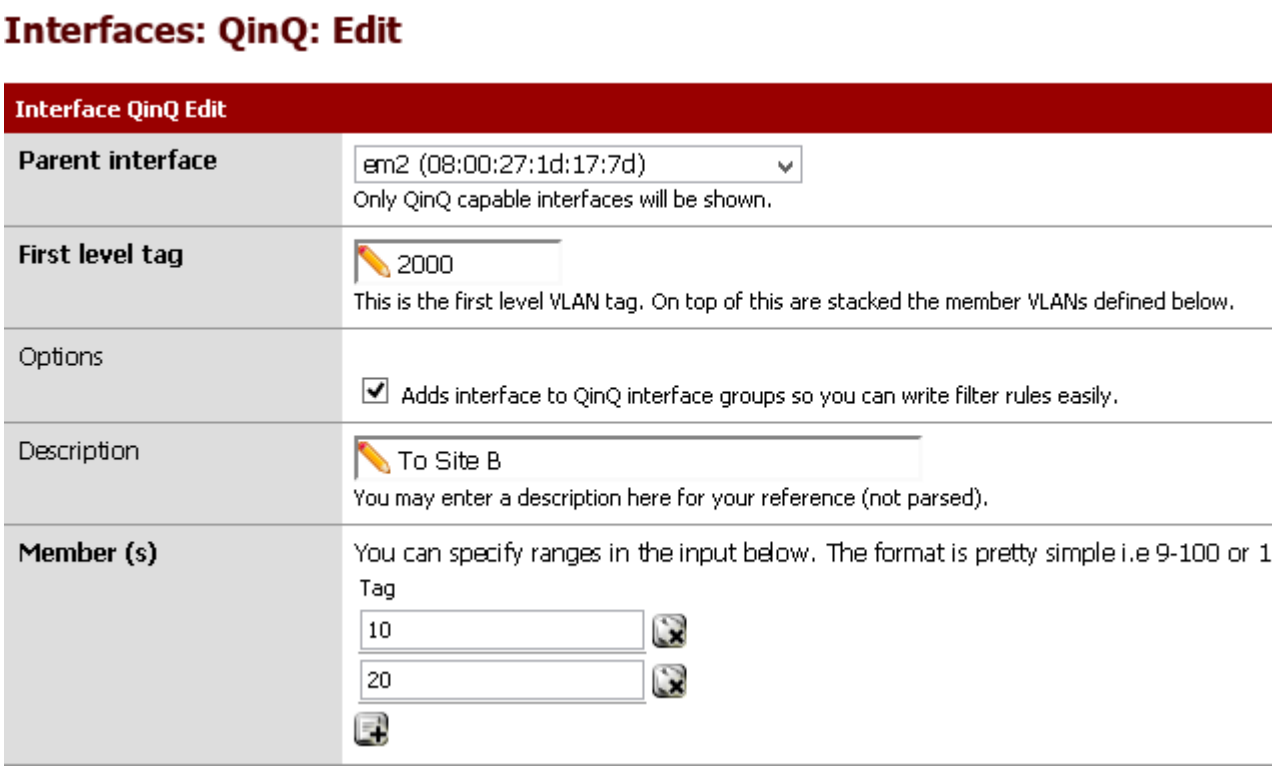
The Description is for your reference, and is optional.

Lastly, you can enter the Member(s) for the QinQ tagging. These can be entered one per row by clicking , or in ranges such as 100-150.

Click Save to complete the interface.

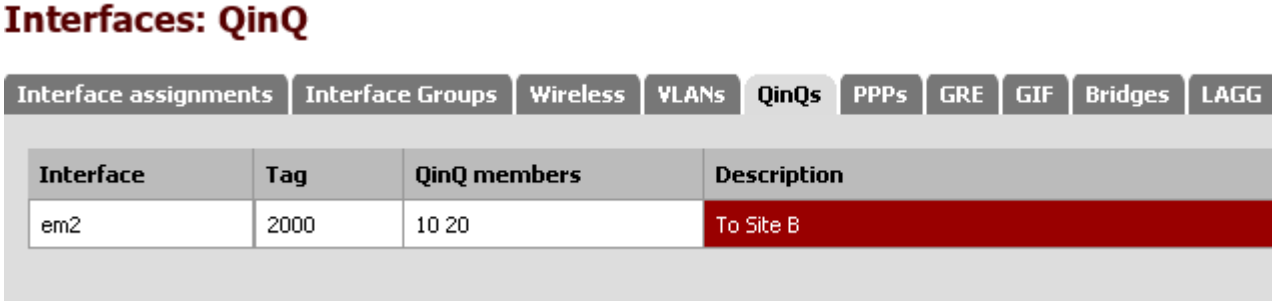
In the following example (Figure 14.20, “QinQ Basic Example”), we setup a QinQ interface to carry tagged traffic for VLANs *10* and *20* across the link on em2 with a first level tag of *2000*.

### Figure 14.20. QinQ Basic Example



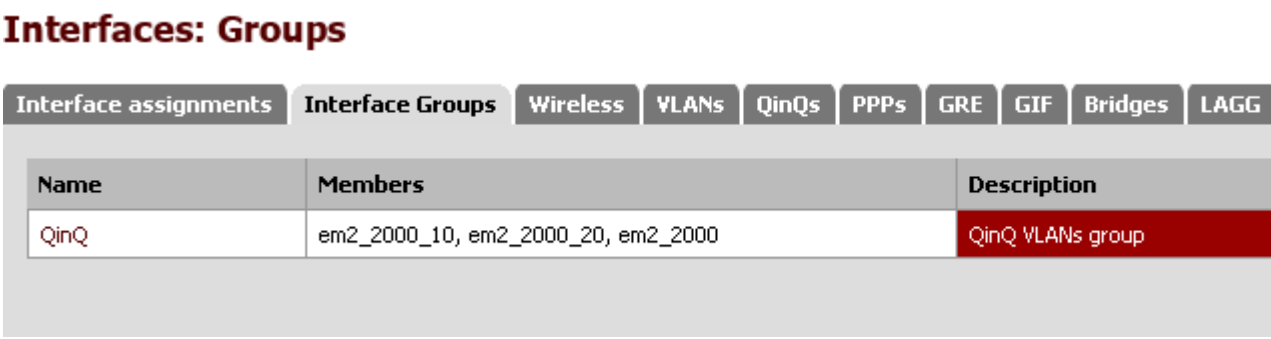

In Figure 14.21, “QinQ List” you can see what this entry looks like on the QinQ tab summary list.

### Figure 14.21. QinQ List




The automatic interface group, shown in Figure 14.22, “QinQ Interface Group”, should not be manually edited. Because these interfaces are not assigned, it's not possible to make alterations to the group without causing problems. If you need to re-create the group, delete it from this list and then edit and save the QinQ instance again to add it back.

**Figure 14.22. QinQ Interface Group**




Now you may add rules to the QinQ tab under Firewall → Rules to pass traffic in both directions across the QinQ links.

From here, how you use the QinQ interfaces is mostly up to you. Most likely, you will want to assign the resulting interfaces and then configure them in some way, or bridge them to their local equivalent VLANs (e.g. bridge an assigned em1_vlan10 to em2_2000_10 and so on).

The QinQ configuration should 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 em2_2000_10 will go through VLAN 2000 on em2, come out the other side on VLAN 2000 on em2 at Site B, and then in em2_2000_10 at Site B.

# Chapter 15. Multiple WAN Connections

The multiple WAN (multi-WAN) capabilities of pfSense allow you to utilize multiple Internet connections to achieve higher uptime and greater throughput capacity. Before proceeding with a multi-WAN configuration, you need a working two interface (LAN and WAN) configuration. pfSense is capable of handling many WAN interfaces, with multiple deployments using 10-12 WANs in production. It should scale even higher than that, though we aren't aware of any installations using more than 12 WANs.

As of pfSense 2.0, all WANs are treated identically in the GUI. Anything you can do with the primary WAN can also be done with OPT WAN interfaces, so there are no longer any significant differences between the primary WAN and additional WANs.

This chapter starts by covering things you should consider when implementing any multi-WAN solution, then covers multi-WAN configuration with pfSense.

# Choosing your Internet Connectivity

The ideal choice of Internet connectivity will depend largely upon the options available at your 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 your connectivity choices for a multi-WAN deployment utilize disparate cabling paths. In many locations, all T1 and DSL connections as well as any others utilizing copper pairs are carried on a single cable subject to the same cable cut.

If you have one connection coming in over copper pair (T1, DSL, etc.), 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 fixed wireless, and fiber services coming in on a different cable path from your copper services.

You cannot rely upon two connections of the same type 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 do use multiple DSL lines or multiple cable modems, though the only redundancy that typically offers is isolating you from modem or other CPE (Customer Premise Equipment) failure. You should consider multiple connections from the same provider as only a solution for additional bandwidth, as the redundancy such a deployment offers is minimal.

## Paths to the Internet

Another consideration when selecting your Internet connectivity is the path from your connection 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 DSL modem and T1 line 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 you vulnerable to extended outages from cable cuts.

# Better Redundancy, More Bandwidth, Less Money

For many years, T1 service has been the choice for any environment with high availability requirements. Generally the Service Level Agreements (SLA) offered on T1 connections are better than other types of connectivity, and T1s are generally seen as more reliable. But with pfSense's multi-WAN capabilities, you can have more bandwidth and better redundancy for less money in many cases.

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 T1 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 a T1 failure or outage of any single service.

# Multi-WAN Terminology and Concepts

This section covers the terminology and concepts you will need to understand to deploy multi-WAN with pfSense.

## 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 your defined gateways from System → Routing, plus any gateway groups you have defined.

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 are what provide the failover and load balancing functionality in pfSense. They are configured under System → Routing, on the Groups tab. In pfSense 1.2.x these were gateway pools handled in a completely different fashion The new grouping mechanism is a lot more powerful, allowing for many more complex failover and load balancing scenarios. When combined with the new gateway options (the section called “Gateway Settings”), it is also a lot more flexible and easier to fine-tune.

## Failover

Failover refers to the ability to use only one WAN interface, but fail over to another WAN if the preferred WAN fails. If you are looking for a way to failover from one firewall to another, that is in Chapter 25, *Firewall Redundancy / High Availability*.

# Load Balancing

Load balancing refers to the ability to distribute load between multiple WAN interfaces. Note that load balancing and failover are not mutually exclusive. Load balancing automatically also provides failover capabilities, as any interface that is down is removed from the load balancing group.

# Monitor IPs

When configuring failover or load balancing, each gateway is associated with a monitor IP (the section called “Monitor IP”). In a typical configuration pfSense will ping this IP, and if it stops responding, the interface is marked as down. Options on the gateway group will let you select different failure triggers besides packet loss. The other triggers are high latency, a combination of either packet loss or high latency, or 100% packet loss.

## So what constitutes failure?

As you may have guessed, it's 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 the section called “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, by default pfSense will flush all states for connections using that gateway. That mechanism will 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 the section called “Gateway Monitoring”.

# 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.
- Make sure you have at least one DNS server set for each WAN gateway under System → General Setup.
- Use the gateway group on your LAN firewall rules.

The remaining sections of this chapter will cover the finer points of implementing those items to result in a capable multi-WAN system.

# 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 traffic, it can only direct it by the gateway IP of the connection. This is fine in most scenarios. If you have multiple connections on the same network using

the same gateway IP, as is common if you have multiple cable modems, you must use an intermediate NAT device so pfSense sees each WAN gateway as a unique IP.

One exception to this is a PPP type WAN (PPPoE, PPTP, etc). These types of connections are able to handle having the same gateway on multiple interfaces, but you must manually add different monitor IPs for their gateway entries (See the section called “Monitor IP”).

# Multiple PPPoE or PPTP WANs

pfSense 2.0 supports PPPoE and PPTP on an unlimited number of WANs, previous versions only supported PPPoE and PPTP on the primary WAN, and other OPT WANs had to use intermediate NAT devices. If you have multiple PPPoE lines from the same ISP and your ISP supports Multi-Link PPPoE, you may be able to bond your lines into a single aggregate link with the total bandwidth of all lines together in a single WAN as seen by pfSense. Configuration of that scenario can be found in the section called “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 any policy routing you have configured on internal interface rules, but rather follows the system's routing table. Hence static routes are required under some circumstances when using OPT WAN interfaces, otherwise only the WAN interface would be used. In pfSense 2.0 and newer you can use floating rules to apply policy routing to traffic leaving from the firewall itself, though that may also require some extra NAT configuration. This only applies to traffic that is initiated by the firewall. 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 and OPT WAN rules, which ensures the reply traffic is routed back out the correct WAN interface.

## 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 endpoint pointing to the specified WAN's gateway to ensure the firewall sends traffic out the correct interface when it is initiating the connection. For mobile connections, the client always initiates the connection, and the reply traffic is correctly routed by the state table.

On pfSense 2.1, an IPsec tunnel may also be set using a gateway group as its interface for failover. This is discussed further in the section called “Multi-WAN Environments”.

## OpenVPN

OpenVPN multi-WAN capabilities are described in the section called “OpenVPN and Multi-WAN”. Like IPsec, it can use any WAN or a gateway group.

## PPTP Server

The PPTP server is not multi-WAN compatible. It can only be used on the WAN interface with the system's default gateway.

### CARP and multi-WAN

CARP is multi-WAN capable as long as all WAN interfaces use static IPs and you have at least three public IPs per WAN. This is covered in the section called “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 will be covered in more detail later in the section called “Multi-WAN for IPv6”.

# Interface and DNS Configuration

First you need to configure your WAN interfaces and DNS servers.

## Interface Configuration

The WAN interfaces first need to be configured. Setup the primary WAN as previously described in the section called “Setup Wizard”. Then for the OPT WAN interfaces, select the desired type of IP configuration, depending on your Internet connection type. For static IP connections, fill in the IP address and add or select a gateway.

## DNS Server Configuration

You will want to configure pfSense with DNS servers from each WAN connection to ensure it is always able to resolve DNS. This is especially important if your internal network uses pfSense's DNS forwarder for DNS resolution. If you only use one ISP's DNS servers, an outage of that WAN connection will result in a complete Internet outage regardless of your policy routing configuration since DNS will no longer function.

### DNS Servers and Static Routes

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 under System → General Setup, so pfSense uses the correct WAN interface to reach that DNS server. DNS servers that come from dynamic gateways are automatically routed back out the proper path. On that page, for each DNS server listed, select the appropriate gateway from the Use gateway drop-down list. You should have at least one gateway from each WAN where possible.

This is required for two reasons. One, most all ISPs prohibit recursive queries from hosts outside their network, hence you must use the correct WAN interface to access that ISP's DNS server. Secondly, if you lose your primary WAN and do not have a gateway chosen for one of your other DNS servers, you will lose all DNS resolution ability from pfSense itself as all DNS servers will be unreachable when the system's default gateway is unreachable. If you are using pfSense as your DNS server, this will result in a complete failure of DNS for your network.

## 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. He uses pfSense to load balance a large number of internal machines out 10 different connections. For more information on this scale of deployment, see the section called “Multi-WAN on a Stick” about "Multi-WAN on a stick" later in this chapter.

# Multi-WAN Special Cases

Some multi-WAN deployments require workarounds due to limitations in pfSense. This section covers those cases and how to accommodate them.

## Multiple Connections with Same Gateway IP

Because of the way pfSense distributes traffic over multiple Internet connections, if you have multiple Internet connections using the same gateway IP, you will need to insert a NAT device between all but one of those connections. This isn't a great solution, but it is workable. We would like to accommodate this in a future release, but it's very difficult because of the way the underlying software directs traffic when doing policy routing.

# Multi-WAN and NAT

The default NAT rules generated by pfSense will translate any traffic leaving the WAN or an OPT WAN interface to that interface's IP address. In a default two interface LAN and WAN configuration, pfSense will NAT all traffic leaving the WAN interface to the WAN IP address. The addition of OPT WAN interfaces extends this to NAT any traffic leaving an OPT WAN interface to that interface's IP address. This is all handled automatically unless Advanced Outbound NAT is enabled.

The policy routing 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 you require Manual Outbound NAT with multi-WAN, you need to ensure you configure NAT rules for all your WAN 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 to add the port forward on the first WAN connection, then click the  to the right of that entry to add another port forward based on that one. Change the interface to the desired WAN, and click Save.

Thanks to pf's reply-to keyword used on WAN rules, when traffic comes in over a specific WAN interface, the return traffic will go back out the way it came into the firewall. So you can actively use port forwards on all WAN interfaces at the same time, regardless of any failover configuration that may be present. This is especially useful for mail servers, as you can use an address on a secondary WAN as a backup MX, allowing you to receive mail even when the primary line is down. This behavior is configurable, for information on this setting, see the section called “Disable Reply-To”.

## Multi-WAN and 1:1 NAT

1:1 NAT entries are specific to a single WAN 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.

# Load Balancing and Failover

The load balancing functionality in pfSense allows you to distribute traffic 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. 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 you want certain traffic, or all your traffic, to utilize one specific WAN connection unless it is unavailable.

# Configuring a Gateway Group for Load Balancing or Failover

In the pfSense WebGUI, browse to System → Routing. On the Groups tab, click . This will bring you to the Gateway Groups Edit screen. The following sections describe each field on this page.

## Group Name

In the Group Name field, fill in 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.

## Gateway Priority

In the Gateway Priority section you can 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, etc.

### 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 the section called “Weight”.

### Failover

Gateways on a lower numbered 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, you can come up with many complicated scenarios that combine both load balancing and failover. For example, if you have *Gateway A* on Tier 1, and then have *Gateway B* and *Gateway C* on Tier 2, then *Gateway D* on Tier 3, you would get the following behavior: *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.

### Virtual IP

Next to each gateway is a Virtual IP drop-down, if any Virtual IPs exist on the gateway's interface. This setting is used for services such as DynDNS, IPsec, or OpenVPN that support using a gateway group as an interface. The Virtual IP field controls which IP address is used for these services when a specific gateway is active. These virtual IPs must be IP alias or CARP type virtual IPs, as services must bind to them.

If there are no eligible Virtual IPs on an interface or you do not wish to use them, then the drop-down's default value of **`Interface Address`** should be used.

## Trigger Level

The Trigger Level drop-down controls when a gateway is removed from use in a group. There are four different modes that can be used.

### Member Down

When Member Down is used, the gateway would only be removed from use when the gateway is down (100% packet loss). This would catch 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 100% loss.

### Packet Loss

The Packet Loss trigger would only consider a gateway down if the packet loss exceeds the threshold defined on the gateway (See the section called “Packet Loss Thresholds”). Latency would not be considered.

### High Latency

The High Latency trigger would only consider a gateway down if the latency exceeds the threshold defined on the gateway (See the section called “Latency Thresholds”). Packet Loss would not be considered.

### Packet Loss or High Latency

If the Packet Loss or High Latency trigger is in use, then a gateway would be removed from use if either the latency or loss thresholds were exceeded. This is the most useful option for most people, and is the most common choice to use. Unless you have a specific need otherwise, use this option.

## Description

You may enter a description here for your reference. This field is shown on the Gateway Groups list and status screen, and does not affect functionality of the group. It is optional.

# Problems with Load Balancing

Some websites store session information including your IP address, and if a subsequent connection to that site is routed out a different WAN interface using a different public IP, the website will not function properly. This is pretty rare and only includes a few banks in my experience. 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.

The sticky connections feature of pf is supposed 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 and a given gateway, it is not based off of the destination. So if 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.

# Verifying Functionality

Once your multi-WAN setup has been configured, you will want to verify its functionality. The following sections describe how to test each portion of your multi-WAN configuration.

# Testing Failover

If you have configured failover, you will want to test it after completing your configuration to ensure it functions as you desire. Don't make the mistake of waiting until one of your Internet connections fails to first try out your failover configuration.

Browse to Status → Gateways and ensure all your WAN gateways are show as "Online" under Status, as well as on the Gateway Groups tab. If they do not, verify you are using a proper monitor IP as discussed in the section called “Monitor IP”.

## Creating a WAN Failure

There are a number of ways you can simulate a WAN failure, differing depending on the type of Internet connection being used. For any type, first try unplugging the target WAN interface's Ethernet cable from the firewall.

For cable and DSL connections, you will also want to try powering off your modem, and just unplugging the coax or phone line from the modem. For T1 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 you would not have otherwise noticed until an actual failure. One of the most common is using a monitor IP assigned to your DSL or cable modem (in some circumstances you may not be aware where your gateway IP resides). 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 you told pfSense to monitor, the connection is still up, so it will not fail over even if the connection is actually down. There are other types of failure that can similarly only be detected by testing all the individual possibilities for failure. Should you need to change to a different monitor IP, it can be edited on the gateway entry as covered in the section called “Monitor IP”.

## Verifying Interface Status

After creating a WAN failure, refresh the Status → Gateways screen to check the current status.

# Verifying Load Balancing Functionality

This section describes how to verify the functionality of your load balancing configuration.

## Verifying HTTP Load Balancing

The easiest way to verify a HTTP load balancing configuration is to visit one of the websites that displays the public IP address that is being used to access the site. A page on the pfSense site is available for this purpose [http://pfsense.org/ip.php], and there are also countless other sites that serve the same purpose. Search for "what is my IP address" and you will find numerous websites that will show you what public IP address is making the HTTP request. Most of those sites tend to be full of spammy ads, so we provide several sites that simply tell you your IP address.

HTTP sites for finding your public IP

- http://www.pfsense.org/ip.php
- http://files.pfsense.org/ip.php
- http://cvs.pfsense.org/ip.php
- http://www.bsdperimeter.com/ip.php

HTTPS site for finding your public IP

- https://portal.pfsense.org/ip.php

If you load one of these sites, and refresh your browser a number of times, you should see your IP address changing if your load balancing configuration is correct. Note if you have any other traffic on your network, you likely will not see your IP address change on every page refresh. Refresh the page 20 or 30 times and you should see the IP change at least a few times throughout the test. If the IP never changes, try several different sites, and make sure your 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 and trying multiple web browsers are good things to attempt before troubleshooting your load balancer configuration further. Using **curl**, as described in the section called "Verifying load balancing" is a better alternative 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) allows you to see the network path taken to a given destination. See the section called "Using traceroute" for details on using **traceroute**.

## Using Traffic Graphs

The real time traffic graphs, under Status → Traffic Graph, are useful for showing the real time throughput on your WAN interfaces. You can only show one graph at a time per browser window, but you can open additional windows or tabs in your browser and show all your WAN interfaces simultaneously. The Dashboard feature in pfSense 2.0 and newer (also available as a beta package in 1.2) enables the simultaneous display of multiple traffic graphs on a single page.

The RRD traffic graphs under Status → RRD Graphs are useful for longer-term and historical evaluation of your individual WAN utilization.

### Note

Your bandwidth usage may not be exactly equally distributed, since connections are simply directed on a round robin basis without regard for bandwidth usage.

# Policy Routing, Load Balancing and Failover Strategies

You will need to determine the multi-WAN configuration that best suits the needs of your environment. This section provides some guidance on common 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 you accomplish this. There is, however, one caveat. If you have two 5 Mbps WAN circuits, you 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, you cannot aggregate

the bandwidth of two Internet connections into a single large "pipe" without involvement from the ISP. With load balancing, since individual connections are balanced in a round robin fashion, you can achieve 10 Mbps of throughput using two 5 Mbps circuits, just not with a single connection. 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. The exception to this is Multi-Link PPPoE (MLPPP), which can achieve full aggregate bandwidth of all circuits in a bundle, but requires special support from the ISP. For more on MLPPP, see the section called “Multi-Link PPPoE (MLPPP)”

In networks with numerous internal machines accessing the Internet, load balancing will enable you to achieve near the aggregate throughput by balancing the many internal connections out all of the WAN interfaces.

# Segregation of Priority Services

In some situations, you 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, you can segregate services 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, some 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. You can setup your policy routing rules in such a way as 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 some scenarios where you may want to only use failover. 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 allow you to achieve this.

Another usage for failover is when you want to ensure a certain protocol or destination always uses only one WAN unless it goes down.

# Unequal Cost Load Balancing

In pfSense 2.0 you can achieve unequal cost load balancing by setting appropriate weights on the gateways as discussed in the section called “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 5, allowing

**Table 15.1. 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% |

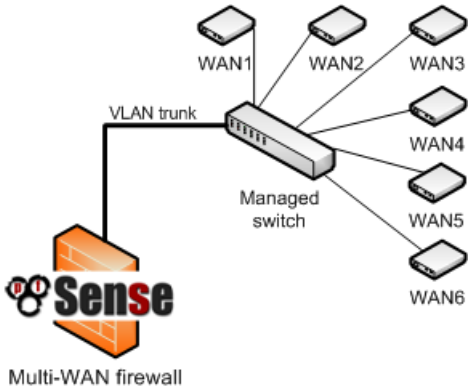
Note that this distribution is strictly balancing the number of connections, it does not take interface throughput into account. This means your bandwidth usage will not necessary be distributed equally, though in most environments it works out to be roughly distributed as configured over time. This also

means if an interface is loaded to its capacity with a single high throughput connection, additional connections will still be directed to that interface. Ideally you would want it to distribute connections based on interface weights and the current throughput of the interface. We are looking into options for this ideal scenario for future pfSense releases, though the existing means of load balancing works very well for most all environments.

# 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. Expanding upon this, we can have multi-WAN on a stick 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 WANs all reside on one physical interface on the firewall, with the internal network(s) on additional physical interfaces. Figure 15.1, “Multi-WAN on a stick” illustrates this type of deployment.

**Figure 15.1. Multi-WAN on a stick**




# Multi-WAN for Services Running on the Firewall

In pfSense 2.0 and higher, it is now possible to direct traffic from the firewall itself into gateway groups using floating rules, allowing local services to take advantage of failover.

# Multi-WAN for IPv6

With pfSense 2.1 you can do Multi-WAN with IPv6 provided that you have multiple ISPs or tunnels setup and working. See the section called “Connecting with a Tunnel Broker Service” if you need help setting up a tunnel.

Gateway Groups work the same for IPv6 as they do for IPv4, but you cannot mix address families 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 might be your actual interface if you have native connectivity, or a tunnel if you are using a tunnel broker.

## Caveats

Traditionally with IPv6 you do not do NAT, as everything is routed. That's great for connectivity, and for businesses or locations that can afford Provider Independent (PI) address space and a BGP peering. It doesn't work so well in practice for home users.

Network Prefix Translation (NPt) will allow you to use one subnet for your LAN and have full connectivity with that subnet via the WAN that actually routes that subnet, and also have it translated on the additional WANs so it appears to originate there. While not true connectivity for the LAN subnet via that path, it is better than no connectivity at all if your primary WAN is down.

This may not work at all for completely dynamic IPv6 types where the subnet is not static. (DHCP-PD, etc.)

## Requirements

To setup Multi-WAN for IPv6 you need:

- Two WANs, and IPv6 connectivity setup on them.
- Gateways added to System → Routing for both, 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 different is NPt instead of NAT.

First, under System → Routing on the Gateway Groups tab, add Gateway Groups for the V6 gateways, with the tiers setup as desired. This works just like IPv4.

Next, navigate to System → General and ensure you have an IPv6 DNS server set for each IPv6 WAN. Again, just like 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: **`Your LAN IPv6 subnet`**.
- Destination IPv6 Prefix: **`Your second WAN's routed IPv6 subnet`**. Note that this is *not* the /64 of the WAN interface itself -- it is the /64 routed to you on that WAN by the upstream.

What this does is akin to 1:1 NAT for IPv4. As traffic leaves the second WAN, if it's coming from the LAN subnet, it will be translated to the equivalent IP in the other subnet. For example if you have *`2001:xxx:yyy::/64`* on your LAN, and *`2001:aaa:bbb::/64`* on your 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 the section called “IPv6 Network Prefix Translation (NPt)”.

As with IPv4 you need to use the Gateway Groups on your LAN firewall rules. Edit your LAN rules for IPv6 traffic and make 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 may prefer to use a "private" IPv6 subnet on their LAN 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 actually bond together multiple PPPoE lines from the same ISP. This means you can get the true aggregate bandwidth of all circuits in the

bundle. For example, if you have three 5 Mbit/s DSL lines in a bundle, you could get 15Mbit/s from a single connection on the line.

## Requirements

The largest hurdle for MLPPP is that your ISP must support it on your circuits. Few ISPs are willing to support MLPPP, so if you can locate one 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 actually fairly simple. Setup a WAN for a single line with your credentials. Once that is setup, navigate to Interfaces → Assign on the PPPs tab. From there, click to edit the entry for your PPPoE WAN, and simply ctrl-click to select the other physical interfaces that belong to the same MLPPP bundle. Save, then apply. pfSense will then attempt to bond the lines using MLPPP.

## Caveats

One downside to using MLPPP is that you can no longer get individual statistics or status for a single line. You have to read through the PPP log in order to determine if one of the links is up or down, as there is not yet a way to query the lines individually. In some cases it's obvious if a line is down, as you may notice that the modem is out of sync or that your maximum attainable bandwidth is reduced, but it makes troubleshooting much more difficult.

# Troubleshooting

This section describes some of the most common problems with multi-WAN and how to troubleshoot them.

## Verify your rule configuration

The most common error when configuring multi-WAN is improper firewall rule configuration. Remember the first matching rule wins — any further rules are ignored. If you add a policy routing rule below the default LAN rule, no traffic will ever match that rule because it will match the default LAN rule first.

If your rule ordering and configuration appears correct, it may help to enable logging on the rules. See the troubleshooting section in the firewall chapter for more information. Ensure the appropriate policy routing rule is passing the traffic.

## Load balancing not working

First, ensure the firewall rule being matched directs traffic to the load balancing gateway group. If the rules are correct, and the traffic is matching a rule with the load balancer gateway group specified, verify that all of the connections show as Online under Status → Gateways. Connections marked as Offline will not be used. Lastly, this may be a problem not with the configuration, but with the testing methodology. Rather than testing with a web browser, try testing with **curl** as described in the section called “Verifying load balancing”.

## Failover not working

If problems occur when an Internet connection fails, typically it's because the monitor IP is still answering so the firewall thinks the connection is still available. Check Status → Gateways to verify. You may be using your modem's IP address as a monitor IP, which will typically still be accessible even if the Internet connection is down.

# Policy routing does not work for web traffic, or appears to not work at all

If you are using a proxy package such as squid that can transparently capture HTTP traffic, this overrides any policy routes that you define for client traffic on that port. So no matter which gateway you set in your firewall rules, traffic for HTTP (TCP port 80) will still go through squid and follow the firewall's default route.

# Chapter 16. Virtual Private Networks

VPNs provide a means of tunneling traffic through an encrypted connection, preventing it from being seen or modified in transit. pfSense offers four VPN options with IPsec, OpenVPN PPTP, 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 your environment. Subsequent chapters go on to discuss each VPN option in detail.

We provide information about PPTP mostly for legacy reasons; PPTP *should not be used* under any circumstances because it is no longer secure. The protocol has been completely compromised, making it so PPTP traffic can be decrypted in every case. More details about this can be found in Chapter 19, *PPTP VPN*.

L2TP on its own is only a tunneling protocol and does not offer any encryption of its own, so it should only be used with that in mind. Because of this, it doesn't fit in with most of the discussion in this chapter. See Chapter 20, *L2TP VPN* for more information on L2TP.

# 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 your network to another location such as a co-location 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

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 T1 circuit has end to end latency of about 3-5 ms, while the latency to the first hop on your ISP's network will generally be at least that much if not higher. Metro Ethernet services have end to end latency of about 1-3 ms, usually less than the latency to the first hop of your ISP's 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. You can minimize latency and maximize VPN performance by using the same ISP for all your 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 your 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.

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 your 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. I have worked in thin client environments that accepted the performance hit, and in others where it was considered unacceptable.

There may be other network applications in your environment that are latency sensitive, where the reduced performance of a VPN is unacceptable. Or you may have all your locations within a relatively small geographical area using the same ISP, where the performance of your VPN rivals that of private WAN connections. Performance is an important consideration when planning a VPN solution.

## Remote access

Remote access VPNs enable users to securely connect into your 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 your network. With the proliferation of smart phones, we also see quite a few people wanting a VPN 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 your wireless networks. This protection is two-fold, in that it provides an additional layer of encryption for traffic traversing your 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 the section called “Additional protection for your 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 you push all your Internet traffic over the VPN, and out to the Internet from your VPN server. This protects you from a number of attacks that people might be attempting on untrusted networks, though it does have a performance impact since it adds additional hops and latency to all your connections. That impact is usually minimal with high speed connectivity when you are geographically relatively close.

# Choosing a VPN solution for your environment

Each VPN solution has its pros and cons. This section will cover the primary considerations in choosing a VPN solution, providing the information you will need to make a choice for your environment.

## Interoperability

If you need a solution to interoperate with a firewall or router product from another vendor, IPsec will usually be the best choice since it is included with every VPN-capable device. It also keeps you from 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 PPTP since it can't be used for site to site connections.

# Authentication considerations

In pfSense 2.x, all of the VPN types can 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 SSL, TLS enabled, 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 your 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 your VPN until that loss is discovered and the certificate revoked.

However 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 you have 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 you have, 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. PPTP is very simple to configure and is the fastest and easiest to get working, but has considerable drawbacks in other areas, especially security. 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, but we try our best to simplify the process in our GUI now that the certificates are managed on the firewall and the 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 in this chapter. When it comes to IPsec, ease of configuration isn't one of its strengths. In previous versions of pfSense OpenVPN was difficult to configure as well, but in pfSense 2.x OpenVPN is fairly simple to setup and use.

# Multi-WAN capable

If you want your users to have the ability to connect to multiple WAN connections, PPTP is not an option because of the way GRE functions in combination with how pfSense's multi-WAN functions. Both IPsec and OpenVPN can be used with multi-WAN.

# Client availability

For remote access VPNs, the availability of client software is a primary consideration. PPTP is the only option with client support built into most operating systems, but all three options are cross platform compatible. Client software is a program that handles connecting to the VPN and handling any other related tasks like authentication, encrypting, routing, etc.

## IPsec

IPsec clients are available for Windows, Mac OS X, BSD and Linux though they are not included in the OS except for some Linux and BSD distributions. A good free option for Windows is the Shrew

Soft client [http://www.shrew.net/]. Mac OS X includes IPsec support, but no user friendly interface for using it. There are free and commercial options available with a user-friendly GUI.

The Cisco IPsec client included with the iOS devices is fully compatible with pfSense IPsec using xauth, and configuration for the client is covered in the section called “iOS Mobile IPsec”.

Many Android phones also include a compatible IPsec client, which is discussed in the section called “Android Mobile IPsec”.

### 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 devices can use a freely available OpenVPN client that works well and doesn't require rooting the device. That client is covered in the section called “Android 4.x”. Older versions of Android may also be able to use OpenVPN via an alternate non-root client covered in the section called “Android 2.1 through 3.2”. There are other options available if the device is rooted, but that is beyond the scope of this book.

As of late January 2013, iOS also has a native OpenVPN client that works without jailbreaking the device. For more information, see the section called “iOS”.

### PPTP

PPTP clients are included in every Windows version since Windows 95 OSR 2, every Mac OS X release, Apple iOS devices, Android devices, and clients are available for all the BSDs and every major Linux distribution. However, as discussed at the start of this chapter, the use of PPTP should be avoided at all costs.

## Firewall friendliness

VPN protocols can cause difficulties for many firewalls and NAT devices. This is primarily relevant to remote access connectivity, where your users will be behind a myriad of firewalls mostly outside of your control 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-T to function, which encapsulates the ESP traffic over UDP port 4500. pfSense does support NAT-T in pfSense 2.0 and later.

### 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. You may want to use a common port like UDP 53 (usually DNS), or TCP 80 (usually HTTP) or 443 (usually HTTPS) or to evade most egress filtering.

### PPTP

PPTP relies on a control channel running on TCP port 1723 and uses the GRE protocol to transmit data. GRE is frequently blocked or broken by firewalls and NAT devices. It is also subject to NAT limitations on many firewalls including pfSense (described in the section called “PPTP Limitations”). PPTP works in many environments, but your users will likely encounter locations where it does not

work. In some cases this can be a significant problem preventing the use of PPTP. As one example, some 3G wireless data providers assign private IPs to customers, and do not properly NAT GRE traffic, making the use of PPTP over 3G impossible on some networks. These factors, combined with the inherent security issues with PPTP, are even more reasons to avoid its use.

## Cryptographically secure

One of the critical functions of a VPN is to ensure the confidentiality of the data transmitted. PPTP has suffered from multiple security issues in the past, and has some design flaws that make it a weak VPN solution. Now that it has been completely compromised, it is in no way a secure way to protect traffic. PPTP is still widely used, though whether it should be is a matter of debate. Where ever possible, I recommend not using PPTP. Some deploy it regardless because of the convenience factor.

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.

OpenVPN's encryption is compromised if your PKI or shared keys are disclosed.

## Recap

Table 16.1, “Features and Characteristics by VPN Type” shows an overview of the considerations provided in this section.

**Table 16.1. Features and Characteristics by VPN Type**

| VPN Type | Client included in most OSes | Widely interoperable | Multi-WAN | Crypto-graphically secure | Firewall friendly |
|---|---|---|---|---|---|
| **IPsec** | no | yes | yes | yes | no (without NAT-T) |
| **OpenVPN** | no | no | yes | yes | yes |
| **PPTP** | yes | n/a | no | no | most |

# 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, you can disable the addition of those rules by checking Disable all auto-added VPN rules under System → Advanced .

## IPsec

Rules for IPsec traffic coming in to the specified WAN interface is automatically allowed as described in the section called “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, but it does automatically add rules permitting traffic from authenticated clients, opposite of the behavior of IPsec and PPTP. Traffic encapsulated within an active OpenVPN connection is controlled via user defined rules on the OpenVPNtab under Firewall → Rules .

## PPTP

PPTP automatically adds rules permitting TCP 1723 and GRE traffic into the WAN IP. Traffic from connected PPTP clients is controlled via user defined rules on the PPTP tab under Firewall → Rules , similar to IPsec.

# VPNs and IPv6

There are some special considerations for VPNs when using them in combination with IPv6. The two main things are whether or not a certain VPN type supports IPv6, and 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 2.1 IPsec supports IPv6 with one quirk — if you use IPv6 peer addresses, the tunnel can only carry IPv6 phase 2 networks, and the same for IPv4. You cannot mix traffic from address families. See the section called “IPsec and IPv6”.

### OpenVPN

OpenVPN fully supports IPv6 for site to site and mobile clients, and tunnels can carry both IPv4 and IPv6 traffic concurrently. See the section called “OpenVPN and IPv6”.

### PPTP

PPTP does not support IPv6, and there are no plans to attempt to do so in the future.

## IPv6 VPN and Firewall Rules

As mentioned briefly in the section called “Firewall and VPN Concerns”, some special care should 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 you have connectivity, if any traffic were intercepted in between the two networks and that traffic was using an unencrypted protocol like HTTP, then it could be compromising your network.

The best 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. You may even want to add an explicit block rule to the top of the WAN rules to ensure that this traffic cannot enter from the WAN directly.

Another less obvious consequence of having dual stack connectivity between your networks is that differences in DNS can cause unintended routing to take place. Suppose you have IPv4 VPN connectivity between two sites, but you have no IPv6 VPN, just standard IPv6 connectivity at both locations. If your local host is set to prefer IPv6, and it receives a AAAA DNS response with the IPv6 IP 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 your DNS does not contain conflicting records,

or that you add floating rules to prevent this IPv6 traffic from going out WAN if you have a VPN in place. 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 [http://tools.ietf.org/html/draft-gont-opsec-vpn-leakages-00].

# Chapter 17. IPsec

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.

For general discussion of the various types of VPNs available in pfSense and their pros and cons, see Chapter 16, *Virtual Private Networks*.

The GUI for IPsec was completely redesigned between pfSense 1.2.3 and 2.x. It now support multiple phase 2 definitions for each tunnel, as well as NAT traversal, IPsec+NAT, and a larger number of encryption and hash options, and many more options for mobile clients, including xauth.

# IPsec Terminology

Before delving too deeply into configuration, there are some terms that are used throughout the chapter that need some prior explanation. Other terms are explained in more detail upon their use in configuration options.

## Security Association

A Security Association (SA) is a one-way tunnel through which encrypted traffic will travel. Each active IPsec tunnel will have two security associations, one for each direction. The 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 manges 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.

## 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 the endpoints using Internet Security Association and Key Management Protocol (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 how it will be encrypted.

## 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 is the part that builds the actual tunnel to be used for transferring data between the endpoints and clients whose traffic is handled by those routers. If phase 2 has been successfully established, the tunnel will be up and ready for use for traffic matching that phase 2 definition.

# IPsec and IPv6

IPsec is capable of connecting to a tunnel over IPv4 or IPv6 phase 1 peer addresses, but 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, if you want to use IPv6 traffic inside the tunnel, then the tunnel must be connected between IPv6 peer IPs, not IPv4. In other words, the inner and outer address family must match, they cannot be mixed. Mobile IPsec clients do not yet support IPv6.

# Choosing configuration options

IPsec offers numerous configuration options, affecting the performance and security of your IPsec connections. Realistically, it matters little which options you choose here as long as you don't use DES, and use a strong pre-shared key, unless you're protecting something so valuable that an adversary with many millions of dollars worth of processing power is willing to devote it to breaking your IPsec. Even in that case, there is likely an easier and much cheaper way to break into your network and achieve the same end result (social engineering, for one).

## 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.

### 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, you may want to use that instead. Whichever protocol is chosen here will be used to validate the Remote Gateway and the associated identifiers.

### 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

In CARP environments (Chapter 25, *Firewall Redundancy / High Availability*), any CARP virtual IP addresses are also available in the Interface drop-down menu. You should choose the appropriate CARP address for your WAN or wherever the IPsec tunnel will terminate on the pfSense system. By using the CARP IP address, it ensures that the IPsec tunnel will be handled by the MASTER member of the CARP cluster, so even if the main firewall is down, the tunnel will connect to whichever CARP cluster member has taken over the MASTER role.

#### IP Alias VIP

If you have multiple IP addresses on an interface using IP Alias VIPs, they will also be available in this list. If you wish to use one of those IPs for the VPN instead, select it here.

#### Multi-WAN Environments

When using Multi-WAN (Chapter 15, *Multiple WAN Connections*), you should pick the appropriate Interface choice for the WAN-type interface to which the tunnel will connect. If you expect the

connection to enter via WAN, pick WAN. If the tunnel should 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.

Starting with pfSense 2.1 you may also choose a gateway group 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 far side router is one that does not support multiple peer IPs, such as another pfSense unit, you will need to combine this with a DynDNS host set using the same gateway group for failover. The DynDNS host will update the IP seen by the far side, so that the remote router will know to accept traffic from the newly activated WAN.

### Wireless Internal Protection

If you are configuring IPsec to add encryption to a wireless network, as described in the section called “Additional protection with VPN”, you should choose the OPT interface which corresponds to your wireless card. If you are using an external wireless access point, pick the interface pfSense can use to connect to the wireless access point.

## Remote Gateway

The Remote Gateway is the IPsec peer for this phase 1. This is the router on the other side of the tunnel to which IPsec will negotiate this phase 1.

## 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 lets you choose 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 we recommend a long/complex key when using PSK mode.

### Mutual RSA

In **`Mutual RSA`** mode, you 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. You must create or import the CA and certificate for the tunnel before attempting to setup the phase 1.

### Mutual PSK+Xauth

Used with mobile IPsec, this selection enables xauth username and password verification along with a shared (or "group") pre-shared key.

### Mutual RSA+Xauth

Used with mobile IPsec, 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, 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.

## Negotiation Mode

Three Negotiation Mode choices are supported: **`main`**, **`aggressive`**, and **`base`**.

### 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, the identifier must be the remote side's IP address and not a custom identifier.

### 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.

### Base Mode

**`Base`** mode is discussed in an IETF draft [http://tools.ietf.org/html/draft-ietf-ipsec-ike-base-mode-02] and is meant to resolve issues with both **`Main`** and **`Aggressive`** modes; It allows for custom identifiers, and is still secure. Support for **`Base`** mode is not as common as the other choices, however.

## My identifier / Peer Identifier

In **`Aggressive`** and **`Base`** modes, you can 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 for you 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 your 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 lets you enter a different IP address to be used as your identifier. One potential use for this would be if your firewall is behind a router performing NAT. You could use the external IP address 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

A string of your choosing to use as the identifier.

### Dynamic DNS

A hostname to resolve and use as the identifier. This is mostly useful if your 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 you can simply use the hostname in the Remote Gateway field and use **`Peer IP Address`** for the identifier.

## Pre-Shared Key (If using Mutual PSK)

This field is used to enter the PSK for phase 1 authentication. As mentioned previously, this should be long/complex key. If this PSK has been provided by your peer, enter it here. If you must generate one, we recommend using a password generation tool set to a length of at least 15, but it can be much longer.

## Policy Generation

This directive controls how pfSense will act as a responder for phase 2 policies, and is not used at all when this firewall initiates the IPsec connection. If there are no existing phase 2 policies and this is set to **`On`**, then pfSense will accept the first policy given by the remote peer. The **`Require`** option is equivalent to **`On`**. **`Unique`** will make pfSense create and track unique policies for each client. **`Off`** prevent policies from being generated automatically, instead relying only on the policies configured manually in phase 2.

The default value of this is **`Off`** for normal tunnels, **`On`** for PSK-only mobile IPsec, and **`Unique`** for other mobile IPsec types.

## Proposal Checking

This setting controls how pfSense will respond to certain parameters supplied by the remote peer. These parameters are: phase 1 lifetime, phase 2 lifetime, phase 2 key length, and phase 2 PFS.

If **`Obey`** is chosen, the values supplied by the remote peer will be used every time. This option is useful when connecting to third party equipment, especially Cisco equipment, that can have a tendency to send unexpected values for these parameters. The downside of **`Obey`** is that if the remote peer has less secure values for these parameters, the integrity of the tunnel could be compromised compared to the settings you want to be used.

**`Strict`** will ensure that only the values for key length and lifetime supplied are used, unless the peer's values are more secure, in which case those may be used instead. If PFS is enabled on both sides, the PFS value must match. If PFS is not required, it will not be enforced.

**`Claim`** works similarly to **`Strict`** except that it will notify the peer about an adjusted phase 2 lifetime and use its own, if its own is longer.

**`Exact`** will reject anything except an exact match of the values.

By default, pfSense uses a value of **`Claim`** for normal tunnels, and **`Obey`** for mobile IPsec tunnels.

## Encryption algorithms

There are many options for encryption algorithms on both phase 1 and phase 2. DES (Data Encryption Standard) is considered insecure due to its small 56 bit key size, and should never be used unless

you are forced to connect with a remote device that only supports DES. The remaining options are all considered cryptographically secure. Which to choose depends on what device you're connecting to, and the hardware available in your system. When connecting to third party devices, 3DES (also called "Triple DES") is commonly the best choice as it may be the only option the other end supports. For systems without a hardware cryptography accelerator, Blowfish and CAST are the fastest options. When using systems with `glxsb` 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 you to select 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.

## Hash algorithms

Hash algorithms are used with IPsec to verify the authenticity of packet data. MD5, SHA1, SHA256, SHA384, and SHA512 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 [http://tools.ietf.org/rfc/rfc4868.txt] compliant on FreeBSD 8.3 upon which pfSense 2.1 is based, and that is the first version on pfSense where support for those higher-value SHA implementations exists. RFC 4868 compliance breaks compatibility with stacks that implemented draft-ietf-ipsec-ciph-sha-256-00 [http://tools.ietf.org/html/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 [http://lists.freebsd.org/pipermail/svn-src-head/2011-February/025040.html] 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 pretty standard 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 config. Certificates can be imported and managed under System → Cert Manager on the Certificates tab. Choose the certificate you'd like to use for this IPsec phase 1 from the list. The CA for this certificate should match the one chosen in the My Certificate Authority option.

## 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 config. A CA can be imported and managed under System → Cert Manager. Choose the CA you'd like to use for this IPsec phase 1 from the list.

## NAT Traversal

NAT Traversal, also known as NAT-T, can encapsulate the ESP traffic for IPsec inside of UDP packets, to more easily function in the presence of NAT. If your firewall, or the firewall on the other end of the tunnel, will be behind a NAT device, you should set this to **`Enable`**. In cases where you know that both ends of the tunnel are directly connected to a publicly routable IP address, it is best to **`Disable`** this as it can cause problems renegotiating a tunnel when it's not needed. In some cases, especially mobile IPsec clients, you may need to **`Force`** this option on to make sure clients always use NAT-T.

## 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 seconds field controls how often a DPD check is attempted, and the retries 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`* retries 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. Since pfSense 2.0, 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

New in pfSense 2.0, this option allows you to use the traditional tunneling mode of IPsec, or transport mode. Tunnel mode was the only option available on pfSense 1.x. On pfSense 2.1, the tunnel mode was also split to specify 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 get delivered to the other side.

### Transport Mode

**`Transport`** mode will encrypt traffic between the IPs used as the phase 1 endpoints. This mode allows encrypting traffic from the firewall's external IP to the far side router's external IP. 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. You cannot set a Local Network or Remote Network 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 your 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 you to enter an IP address or subnet manually.

### NAT Address for NAT+IPsec

If you need to perform NAT on your local IPs to make them appear as a different subnet, or one of your public IPs, you may do so using the NAT fields underneath Local Network. If you specify a single IP address in Local Network and a single IP address in the NAT field, then a 1:1 NAT rule will be added between the two. 1:1 NAT is also setup if you use a subnet in both fields. If you use a Local Network that is a subnet, but a NAT address that is a single IP, then a 1:many NAT (PAT) rule is added 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 you do not need to do NAT 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.

## Protocol

With IPsec you have the option of choosing AH (Authenticated Header) or ESP (Encapsulating Security Payload). In nearly all circumstances, you should use ESP, as it is the only option that encrypts traffic. AH only provides assurance the traffic came from the trusted source and is rarely used.

## Encryption algorithms

The phase 2 encryption choices allow for multiple selections. The advice earlier in this chapter, in the section called “Encryption algorithms”, still applies. However, you can select multiple options so that either multiple choices will be accepted when acting as a responder, or multiple combinations will be tries when working as an initiator. It's best to only select the single cipher that you want to use, but in some cases selecting multiple will allow a tunnel to work better in both a responder and initiator role.

## Hash algorithms

As with the Encryption Algorithms, here you can select multiple hashes. We still recommend only selecting the one hash you need. For more discussions on the quality of the various hash types, see the section called “Hash algorithms”.

## 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 the section called “DH key group”).

## Lifetime

The lifetime specifies how often the connection must be rekeyed, specified in seconds. 3600 seconds on phase 2 is a pretty standard configuration and is appropriate for most scenarios.

## Automatically Ping Host (Keep Alive)

For use on non-mobile tunnels, this option will cause the firewall to initiate a ping periodically to the IP specified. This option only works if the firewall has an IP inside of this phase 2's Local Network and the value of the ping host here must be inside of the Remote Network.

# IPsec and firewall rules

When you configure an IPsec tunnel connection, pfSense automatically adds hidden firewall rules to allow UDP ports 500 and 4500, and the ESP protocol from the Remote gateway IP destined to the Interface IP specified in the configuration. When Allow mobile clients 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. If you check that box, you must manually add firewall rules 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, IPsec tab. Here you can restrict what resources can be accessed by 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 allows you to interconnect 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, just 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 even exists. No client software is needed, and all of the tunnel work is handled by the routers on either end of the connection. 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 make sure that both sides have matching settings for authentication, encryption, etc. Before starting, make a note of the local and remote WAN IP addresses, as well as the local and remote internal subnets that you will be connecting. An IP from the remote subnet to ping is optional, but recommended to keep the tunnel alive. The system doesn't check for replies, as any traffic initiated to an IP 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 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.

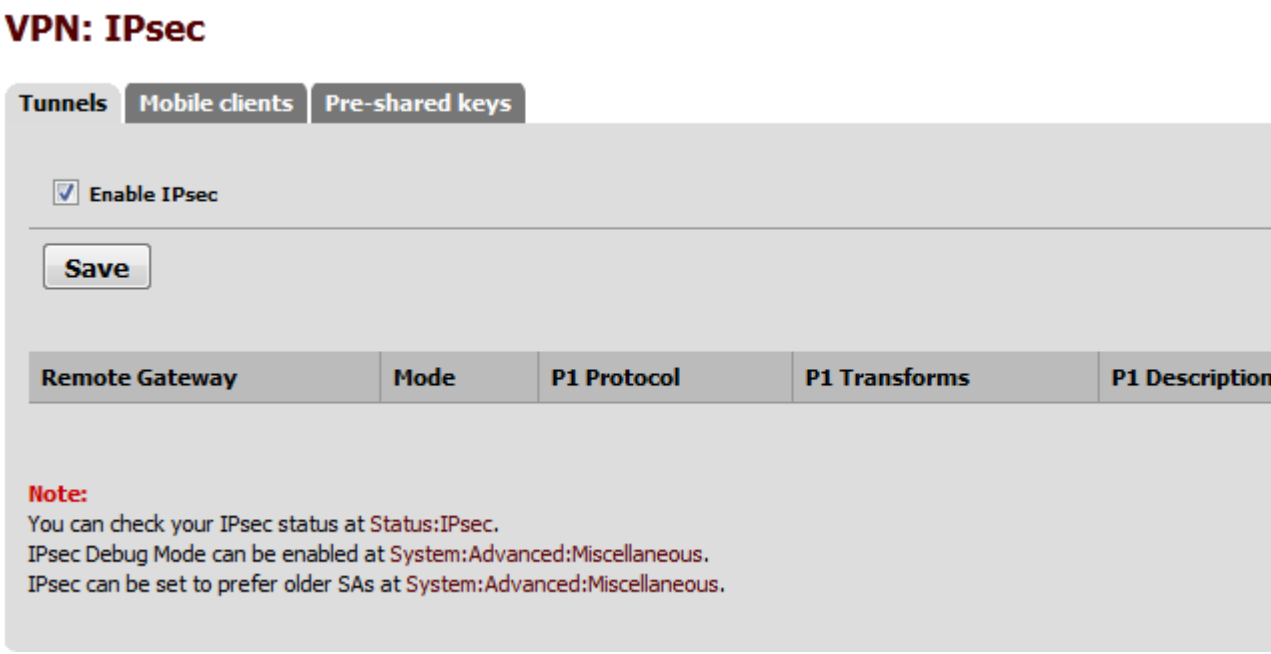
In this example and some of the subsequent examples in this chapter, the following settings will be assumed:

**Table 17.1. IPsec Endpoint Settings**

| **Site A** | | **Site B** | |
|---|---|---|---|
| Name | Louisville Office | Name | London Office |
| WAN IP | 172.23.1.3 | WAN IP | 172.16.1.3 |
| LAN Subnet | 192.168.1.0/24 | LAN Subnet | 10.0.10.0/24 |
| LAN IP | 192.168.1.1 | LAN IP | 10.0.10.1 |

We will start with Site A. First, we must enable IPsec on the router. Navigate to VPN → IPsec, check Enable IPsec, then click Save (Figure 17.1, “Enable IPsec”).

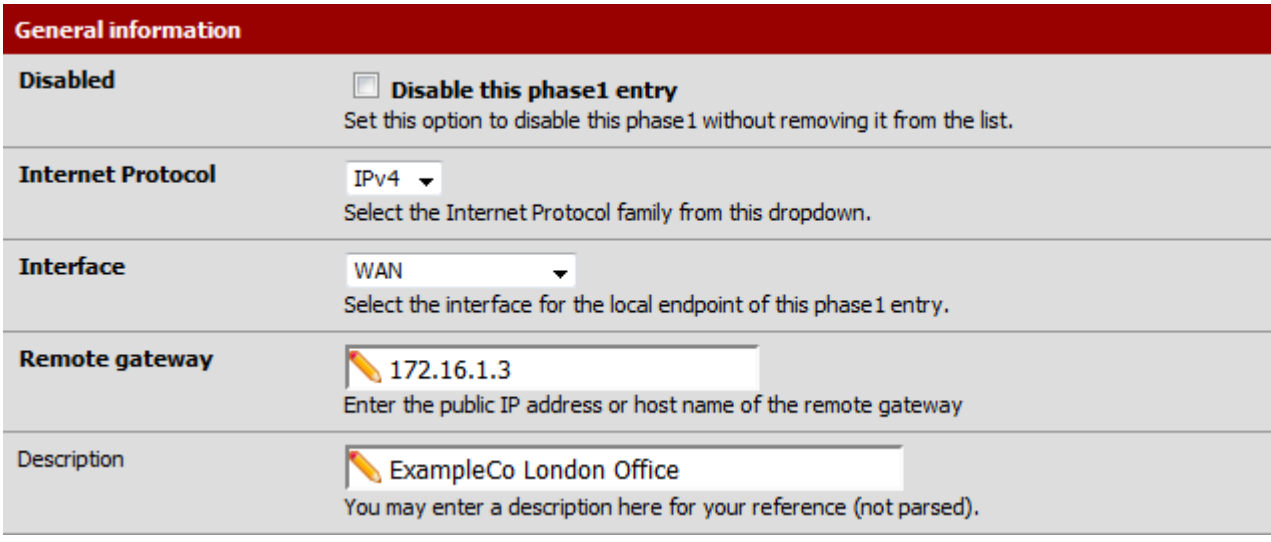
**Figure 17.1. Enable IPsec**




Now, create the tunnel by pressing the button. You will now see a large page that has the phase 1 configuration for the tunnel. Don't be too discouraged, as many of these settings may be left at their default values.

To get started, fill in the top section that holds the general phase 1 information, shown in Figure 17.2, “Site A VPN Tunnel Settings”. Items in bold are required. Items in bold are required. Make sure that the Disable this tunnel box is unchecked. The Internet Protocol should be **`IPv4`**. The interface setting should likely be **`WAN`**, but see the note at the section called “Interface Selection” on selecting the proper interface if you are unsure. The Remote Gateway is the WAN address at Site B, **`172.16.1.3`**. Finally, enter a Description for 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 may also help future administration. We'll put "**`ExampleCo London Office`**" in the description so we have some idea where the tunnel terminates.

**Figure 17.2. Site A VPN Tunnel Settings**

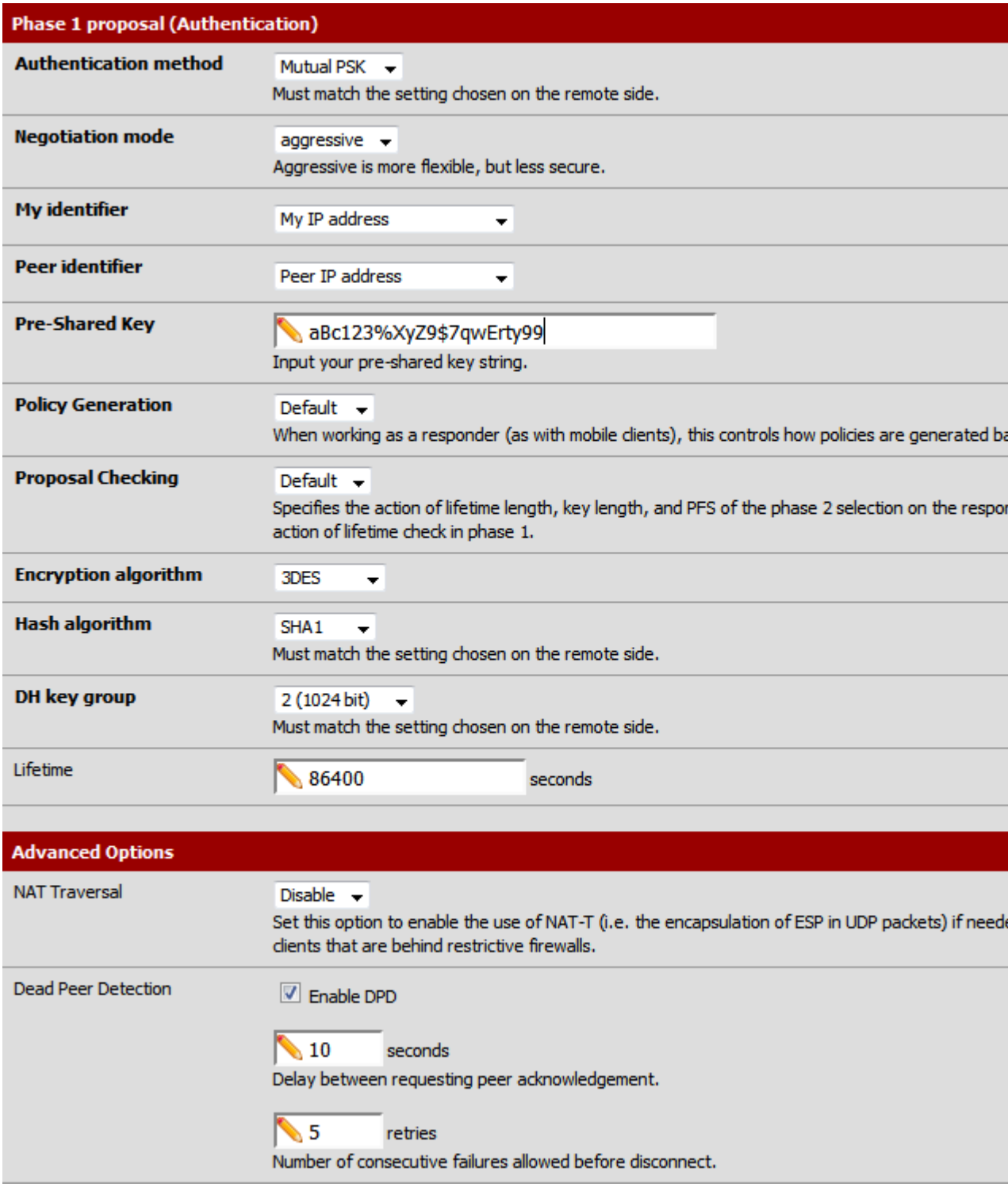



The next section controls IPsec phase 1, or Authentication. It is shown in Figure 17.3, “Site A Phase 1 Settings”. The defaults are desirable for most of these settings, and simplifies the process. The most

important setting to get right is the Pre-Shared Key. 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 you may want to write it down, or copy and paste it elsewhere. Copy and paste may come in handy, especially with a complex key like **`aBc123%XyZ9$7qwErty99`**. A Lifetime setting may also be specified, otherwise the default value of **`86400`** will be used.

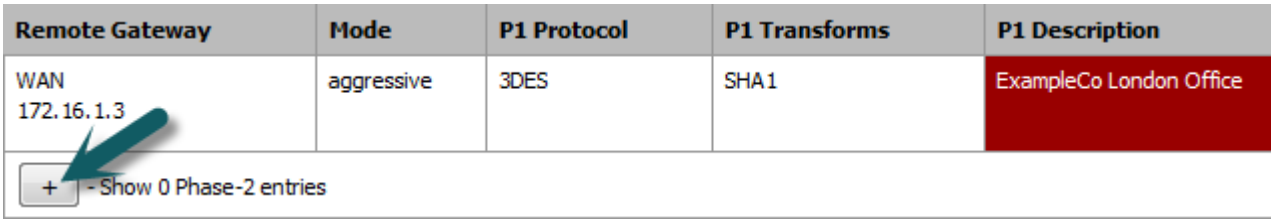
Set NAT Traversal to **`Disable`**, since in this example neither firewall is behind NAT. Check Dead Peer Detection (DPD) and enter reasonable values, such as **`10`** seconds and **`5`** retries. Depending on your needs a higher value may be better, more like **`30`** seconds and **`6`** retries, but a problematic WAN connection on either side might make that too low. Click Save to complete the phase 1 setup.

**Figure 17.3. Site A Phase 1 Settings**

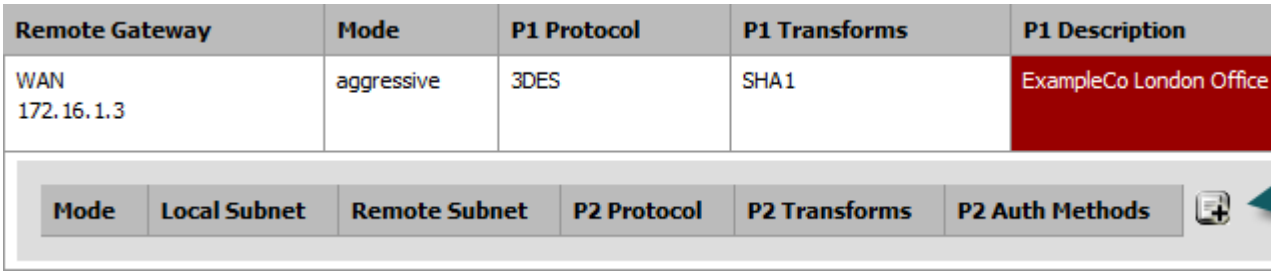



After the phase 1 has been added, you will need to add a new phase 2 definition to the VPN. To do this, click the + button as seen in Figure 17.4, “Site A Phase 2 List (Empty)” to expand the phase 2 list for this VPN. Since there are no phase 2 entries yet, click at the right side to add one, as seen in Figure 17.5, “Adding a Phase 2 entry to Site A”.

### Figure 17.4. Site A Phase 2 List (Empty)

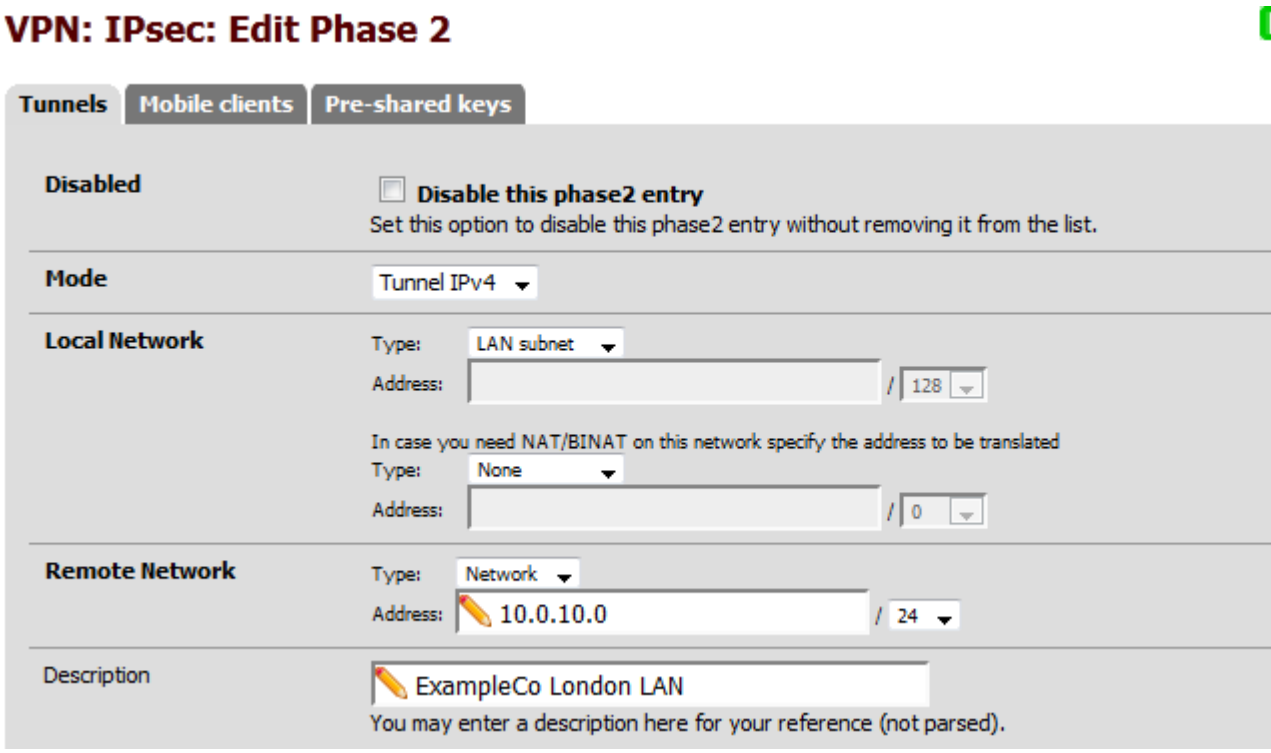



### Figure 17.5. Adding a Phase 2 entry to Site A




Now you can add setting for phase 2 on this VPN. The phase 2 (Figure 17.6, “Site A Phase 2 General Settings”) settings can have a little more variability. In this case, since we want to tunnel traffic, we select **`Tunnel IPv4`** for the Mode. For the Local Subnet, is probably best to leave this as **`LAN Subnet`**. You could also change this to **`Network`** and fill in the proper values, in this case **`192.168.1.0/24`**, but leaving it as **`LAN Subnet`** will ensure that should the network ever be renumbered, this end of the tunnel will follow. Note the other end must be changed manually. The NAT subnet should be **`None`**. The Remote Subnet will be the network at Site B, in this case **`10.0.10.0/24`**.

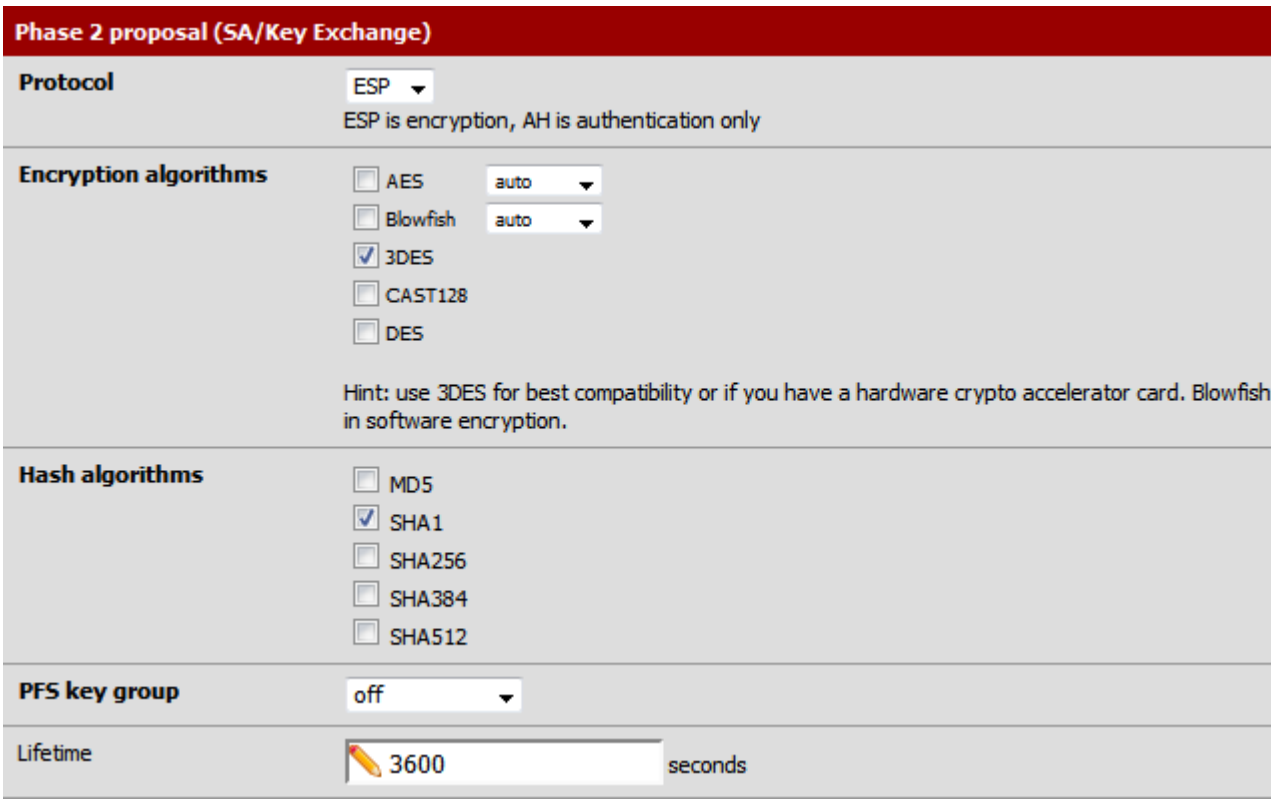
### Figure 17.6. Site A Phase 2 General Settings




The remainder of the phase 2 settings, seen in Figure 17.7, “Site A Phase 2 Settings”, cover the encryption of the traffic. The Protocol choice should **ESP** for encryption. The Encryption algorithms

and Hash algorithms can both be set to allow multiple options, and both sides will negotiate and agree upon the settings. In some cases that may be a good thing, but it is usually better to restrict this to the options that you know will be in use. For this example, the only Encryption algorithm selected is 3DES, and the only Hash algorithm selected is SHA1. PFS, or Perfect Forward Secrecy, can help protect against certain key attacks, but is optional. A Lifetime setting may also be specified, otherwise the default value of **3600** will be used.

### Figure 17.7. Site A Phase 2 Settings

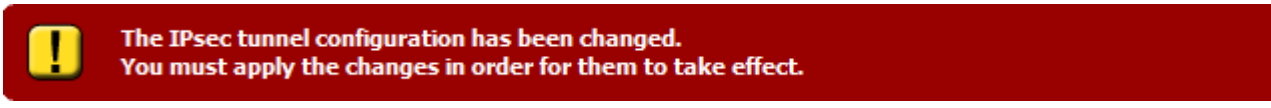



Lastly, you can enter an IP address for a system on the remote LAN that should periodically be sent an ICMP ping, as in Figure 17.8, “Site A Keep Alive”. The return value of the ping is not checked, this will only ensure that some traffic is sent on the tunnel so that it will stay established. In this setup, we can use the LAN IP address of the pfSense router at Site B, **10.0.10.1**.

### Figure 17.8. Site A Keep Alive


Click the Save button, and then you will need to click Apply changes on the IPsec Tunnels screen, as seen in Figure 17.9, “Apply IPsec Settings”.

### Figure 17.9. Apply IPsec Settings



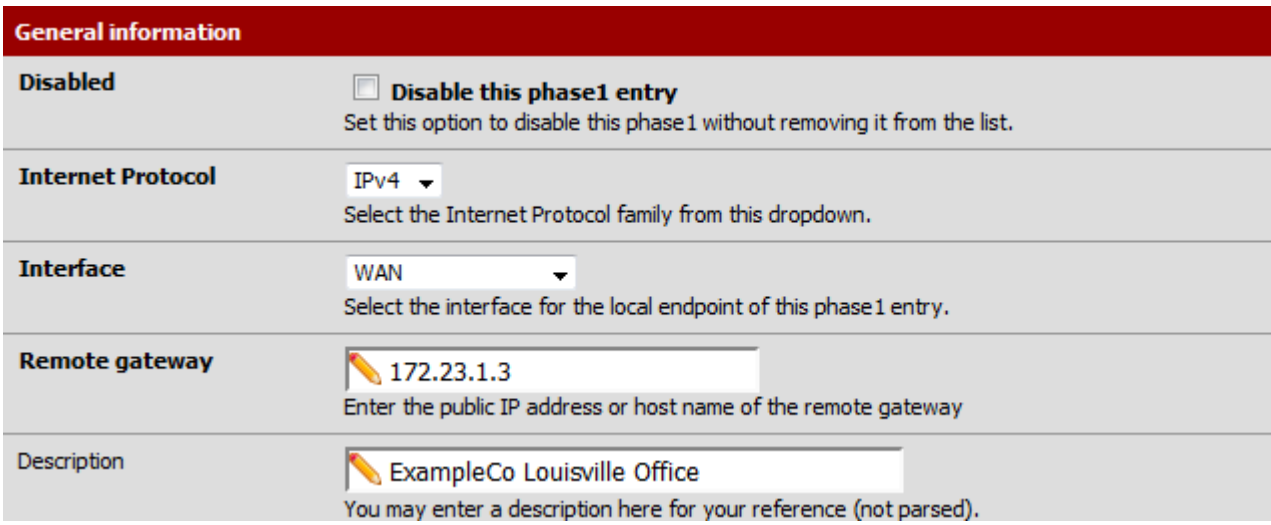

The tunnel for Site A is finished, but now firewall rules are needed to allow traffic from Site B's network to come in via the IPsec tunnel. These rules must be added to the IPsec tab under Firewall →

Rules. See the chapter on Firewall rules for specifics on adding the rules. You may be as permissive as you like, (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 **`10.0.10.0/24`**. The destination addresses should be the Site A network, **`192.168.1.0/24`**.

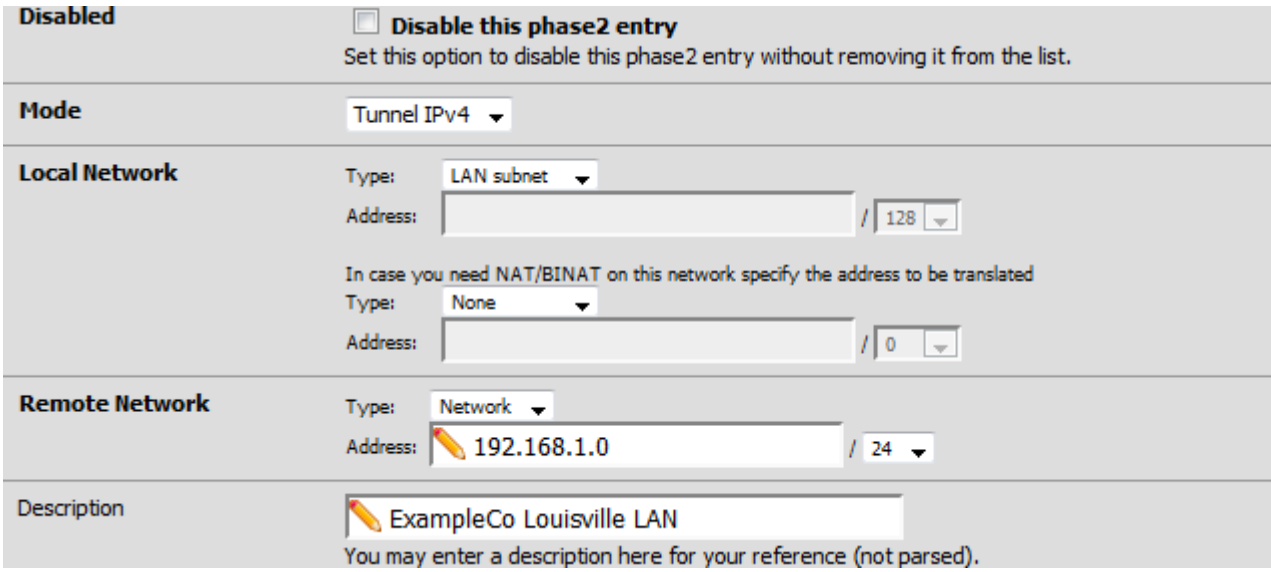
Now that Site A is configured, it is time to tackle Site B. Repeat the process on Site B's router to enable IPsec and add a tunnel.

Only three parts of this setup will differ from Site A. Those are the phase 1 settings, the phase 2 tunnel networks, and the keep alive setting, as you can see in Figure 17.10, “Site B Phase 1 Settings” and Figure 17.11, “Site B Phase 2 Settings”. On phase 1, make sure that the Disable this tunnel box is unchecked. The interface setting should be **`WAN`**. Fill in the Dead Peer Detection (DPD) value with the same setting as Site A. The Remote Gateway is the WAN address at Site A, **`172.23.1.3`**. A Description for the tunnel is still a good idea. We'll put "**`ExampleCo Louisville Office`**" on this side. Click Save and then add a phase 2 to this side. For the Site B phase 2 settings, the Local Subnet, it is probably best to leave this as **`LAN Subnet`**. You could also change this to **`Network`** and fill in the proper values, in this case **`10.0.10.0/24`**. The Remote Subnet will be the network at Site A, in this case **`192.168.1.0/24`**.

### Figure 17.10. Site B Phase 1 Settings




### Figure 17.11. Site B Phase 2 Settings



The phase 1 and phase 2 settings must match Site A exactly. Review that section of this example for the details and figures.

The last change is the keep alive setting (Figure 17.12, “Site B Keep Alive”). In this setup, we can use the LAN IP address of the pfSense router at Site A, **`192.168.1.1`**.

### Figure 17.12. Site B Keep Alive


Now click the Save button, and then click Apply changes on the IPsec Tunnels screen.

As with Site A, you must also add firewall rules 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 the section called “IPsec and firewall rules”. This time, the source of the traffic would be Site A, destination Site B.

Both tunnels are now configured and should be active. Check the IPsec status by visiting Status → IPsec. You should see a description of the tunnel along with an indicator icon for its status.

If you do not see a icon, 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, you will see a connect button on this screen that will initiate a ping to the remote phase 2. Click the button to attempt to bring up the tunnel, as seen in Figure 17.13, “Site A IPsec Status”. If the connect button doesn't 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 the section called “Testing IPsec Connectivity” for other means of testing a tunnel.

### Figure 17.13. Site A IPsec Status


Failing that, the IPsec logs will offer an explanation. They are located under Status → System Logs on the IPsec VPN tab. Be sure to check the status and logs at both sites. For more troubleshooting information, check the the section called “IPsec Troubleshooting” section later in this chapter.

# Routing and gateway considerations

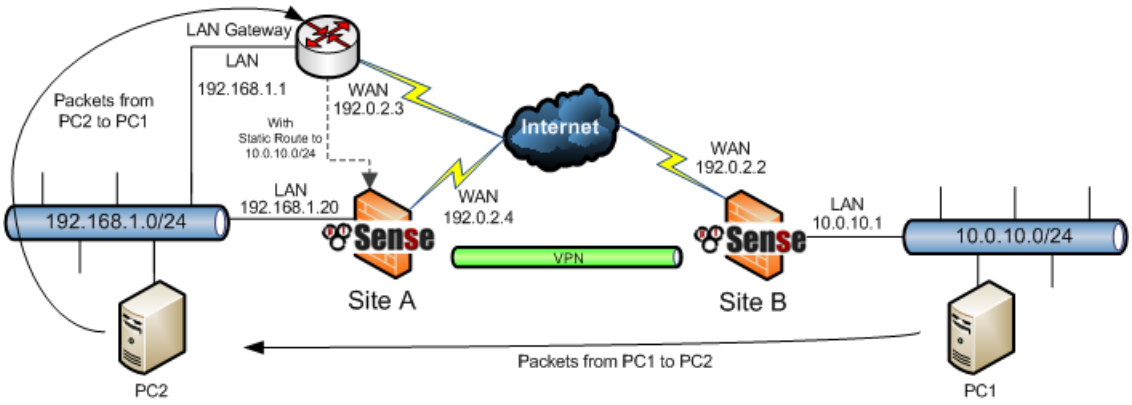
When the VPN endpoint, in this case a pfSense router, is the default gateway for a network there should be no problems with routing. As a client PC sends traffic, it will go to the pfSense box, over the tunnel, and out the other end. However, if the pfSense router is *not* the default gateway for a given network, then other routing measures will need to be taken.

As an example, imagine that the pfSense router is the gateway at Site B, but not Site A, as illustrated in Figure 17.14, “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 Site B's router, across the tunnel, out the pfSense router at Site A, and on to PC2. But what happens on the way back? PC2's gateway is another router entirely. The reply to the ping will be sent to the gateway system 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. First, 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 router. Even with this route, additional complexities are introduced because this scenario results in asymmetric routing as covered in the section called “Bypass Firewall Rules for Traffic on Same Interface”. Should that not work, a static route could be added to the client systems individually so that they know to send that traffic directly

to the pfSense box 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. Last but not least, in some situations it may be easier to make the pfSense box the gateway and let it handle your Internet connection.

**Figure 17.14. Site to Site IPsec Where pfSense is not the Gateway**




# Routing multiple subnets over IPsec

If you need to route multiple IP subnets over IPsec, pfSense 2.0 allows the definition of multiple subnets per IPsec connection by defining a new phase 2 entry for each subnet you want to be able to use the firewall. On the older 1.2.x versions you can still do it, but it's not as convenient. You have two options — CIDR summarization and parallel IPsec tunnels.

### Note

Traffic will traverse an IPsec tunnel only if it matches an existing SAD entry. Static routes will *not* route traffic over an IPsec connection, never configure static routes for any IPsec traffic except in the case of traffic initiated from pfSense itself (which will be discussed later).

## CIDR Summarization

If the subnets are contiguous, you can route multiple subnets on one tunnel using a larger subnet which includes all the smaller subnets. For example if one site includes the subnets 192.168.0.0/24 and 192.168.1.0/24, that can be summarized as 192.168.0.0/23. See the section called “CIDR Summarization” for more information.

## Parallel IPsec Tunnels

The only option on pfSense 1.2.3 if the subnets are not summarized is to create parallel IPsec tunnels, one for each subnet.

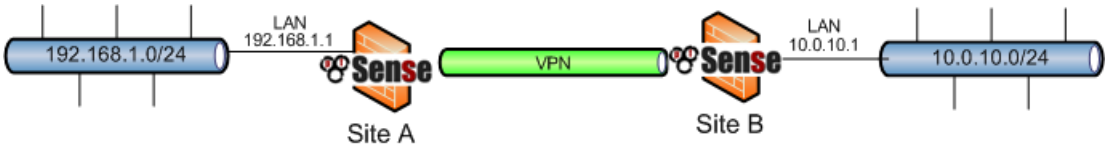
Click the to the right of the first connection to add another based on this one. Change only the remote subnet (to the second subnet you wish to connect) and set the PSK to something different from the first connection. Save your changes.

# pfSense-initiated Traffic and IPsec

To access the remote end of IPsec connections from pfSense itself, you will need to "fake" the system by adding a static route pointing the remote network to the system's LAN IP. Note this example presumes the VPN is connecting the LAN interface on both sides. If your 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's routing table, which will likely send this traffic out your WAN interface rather than over the IPsec tunnel. Take Figure 17.15, “Site to Site IPsec”, for example.

**Figure 17.15. Site to Site IPsec**




You need to add a static route on each firewall, which is done by first adding a gateway pointing to the firewall's LAN IP (See the section called “Gateways”), and then adding a static route using this gateway (See the section called “Static Routes”). Figure 17.16, “Site A — Static route to remote subnet” and Figure 17.17, “Site B — Static route to remote subnet” show the route to be added on each side.

**Figure 17.16. Site A — Static route to remote subnet**

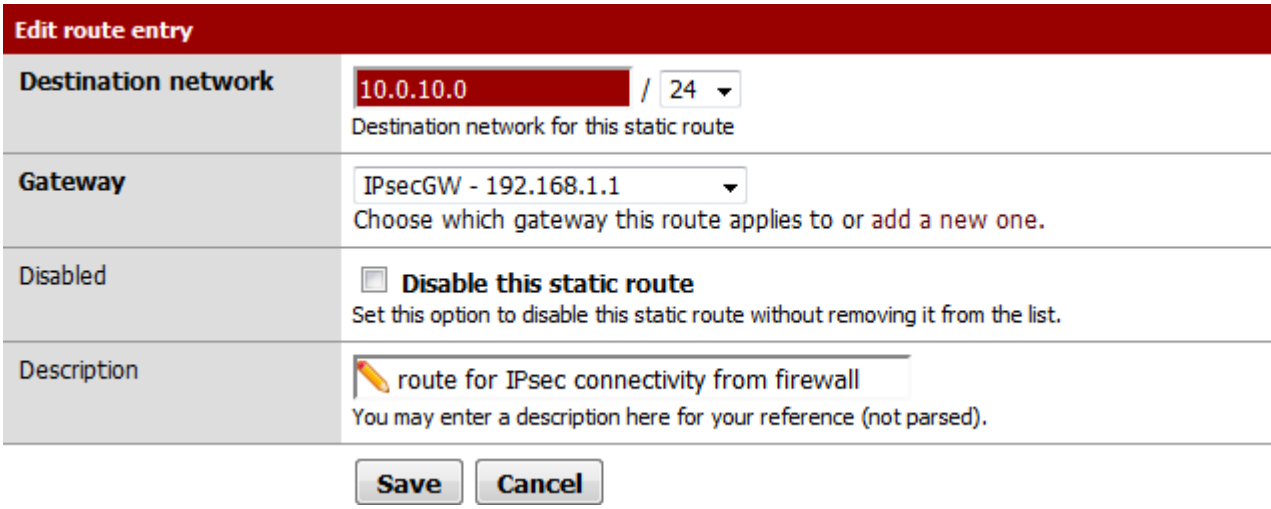



**Figure 17.17. Site B — Static route to remote subnet**


# Mobile IPsec

Mobile IPsec will allow you to make a so-called "Road Warrior" style connection, named after the variable nature of anyone who is not in the office that needs to connect back to the main network.

It may 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 your network, and will introduce routing complications. This is where IPsec Mobile Clients are useful.

There is only one definition for Mobile IPsec on pfSense, so you may be wondering how to setup multiple clients. Instead of relying on a fixed address for the remote end of the tunnel, IPsec authentication via xauth is possible to allow a username/password login to identify a unique user. This allows the clients to be authenticated and distinguished from one another.

Before you begin configuring clients, you will need to choose an IP address range they will be using. Care will be needed to 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, *10.50.99.0/24* will be used, but it can be any unused subnet that you desire.

This is different from the mobile IPsec style suggested for 1.2.3, but using xauth is both more secure and also allows connections from mobile devices running iOS and some versions of Android. Also, pfSense 2.x supports pushing addresses and other settings to the clients for automatic configuration.

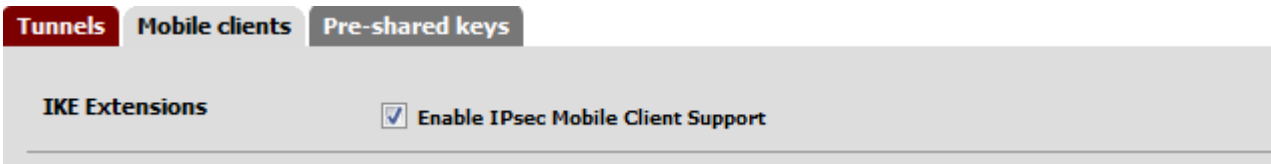
# Example Server Configuration

There are several components to the server configuration for mobile clients: Setting the Mobile Client settings and creating the phase 1 and phase 2 for the client connection, and adding IPsec firewall rules. After that, users must be added with the right permissions to use the VPN.

## Mobile Client Settings

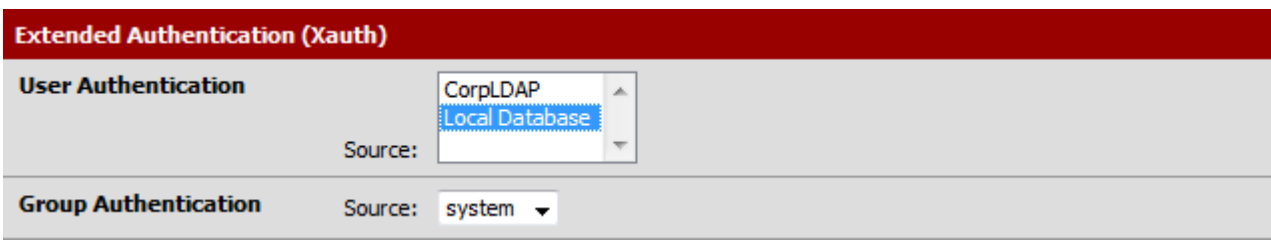
First, we must enable IPsec on the router if you haven't done so already. Navigate to VPN → IPsec, check Enable IPsec, then click Save. With IPsec enabled, mobile client support must also be enabled. From VPN → IPsec, click on the Mobile clients tab (Figure 17.18, “Enable Mobile IPsec Clients”). Check the Enable IPsec Mobile Client Support box, and then continue on to the next set of options.

### Figure 17.18. Enable Mobile IPsec Clients




Since we want xauth and we want to use the pfSense user manager for authentication, we can leave the authentication sources set to **`Local Database`**, as seen in Figure 17.19, “Mobile Clients Authentication”. LDAP and RADIUS servers defined in the User Manager (Chapter 7, *User Management and Authentication*) can also be used for authenticating users.

### Figure 17.19. Mobile Clients Authentication



New in pfSense 2.x is the ability to push settings to the client for things like the client IP and DNS. These options are shown in Figure 17.20, “Mobile Clients Pushed Settings”.

The Virtual Address Pool defines the pool of IPs that will be handed out to clients. We will use *10.50.99.0/24* in this example.

The Network List option 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. We only want the client to reach the network in our phase 2 for this example, so we will check this option.

If Save Xauth Password is checked, clients that support this control will allow the user to save their credentials. This is mainly respected by Cisco-based clients like the one found on iOS and Mac OSX. For this example, we want to allow that behavior, so the box will be checked.

If you check DNS Default Domain and enter a value, then this value 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*.

The Split DNS option 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 our example, we have both *example.com* and *example.org* and want DNS requests for those two domain to go to our servers, so we enter those values here separated by a comma.

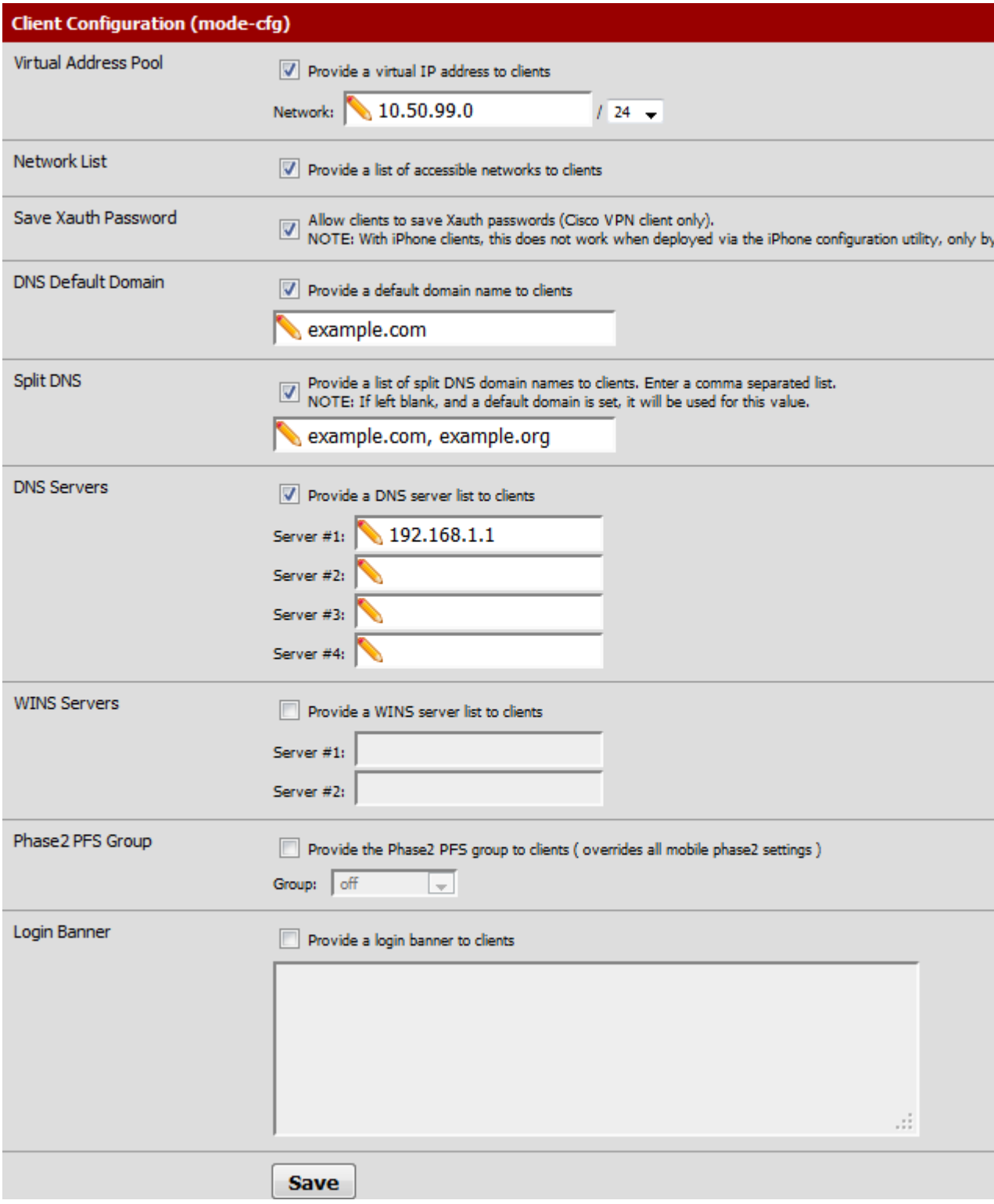
You can supply DNS Servers to clients by checking Provide a DNS server list to clients, and entering IP addresses for the local DNS servers, such as *192.168.1.1*.

## Note

If you intend your mobile clients to route to the Internet over the VPN, you must make sure the clients get a DNS Server from the firewall using this option, and that they do not have Split DNS enabled. If you do not configure it this way, 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.

You can also provide WINS Servers, set the Phase 2 PFS Group, and even show a Login Banner on clients, but for this example we won't be using those, so they are left disabled.

**Figure 17.20. Mobile Clients Pushed Settings**




After saving the settings on the Mobile Clients tab, pfSense will warn you that you don't get have a phase 1 definition for your mobile clients. Press the Create Phase 1 button to make one and start the next step (Figure 17.21, “Mobile Clients Phase 1 Creation Prompt”).

### Figure 17.21. Mobile Clients Phase 1 Creation Prompt

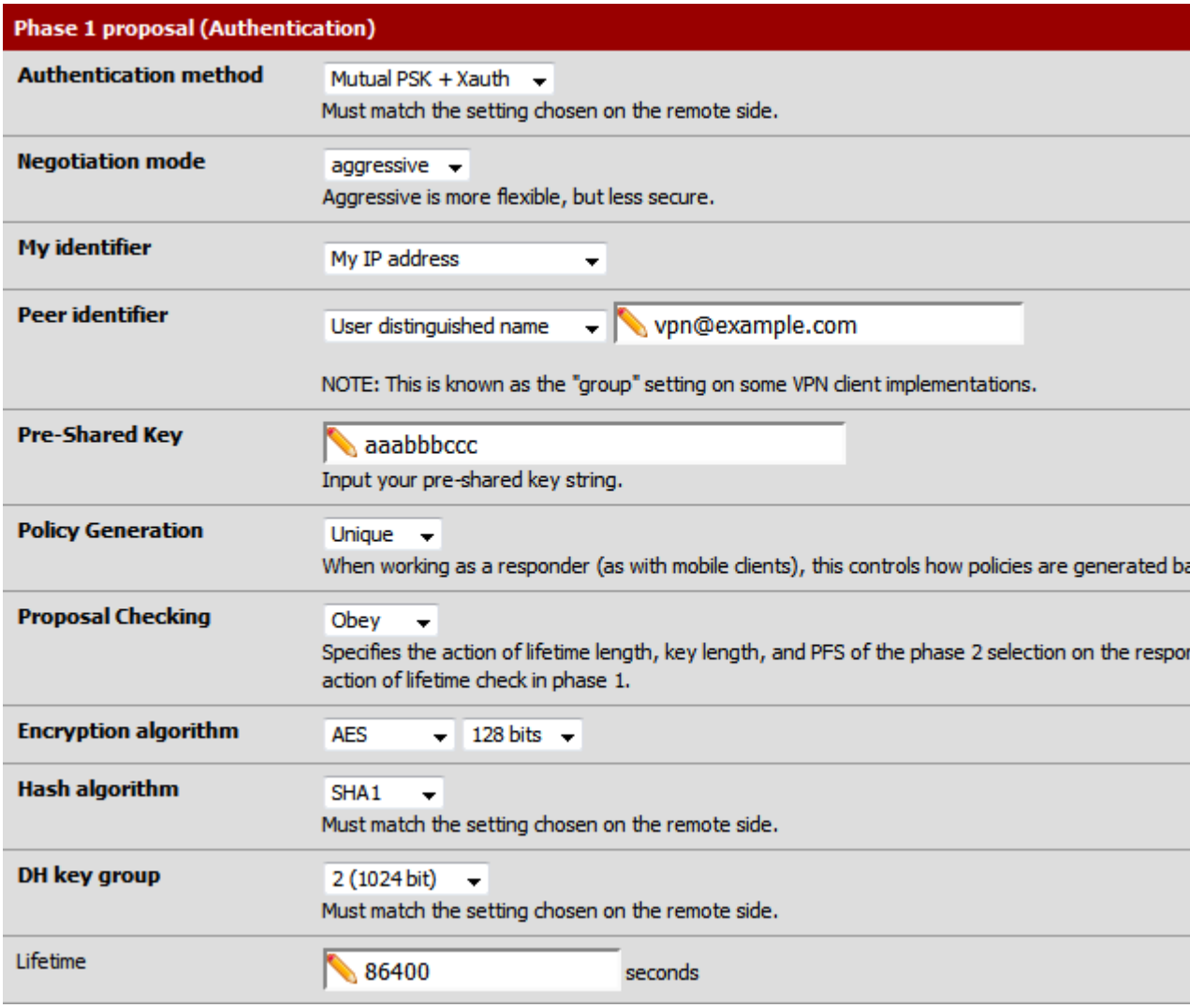

Now we see the phase 1 settings for mobile clients. For starters, the Authentication Method should be set to **Mutual PSK+Xauth** since we intend to use this from Android and iOS. These operating systems have a particular quirk in that they require very specific values for the encryption, hash, lifetimes, etc. It's important that if you want to use mobile IPsec with Android or iOS that you set the values exactly as shown in Figure 17.22, “Mobile Clients Phase 1” and Figure 17.23, “Mobile Clients Phase 2”.

First, set **aggressive** for the Negotiation mode. The clients will be connecting from random/dynamic IPs so this will allow using a custom identifier type for the peer instead of the IP address. Using **My IP address** for the My identifier option is best. The Peer Identifier is also known as the group name in certain client configurations, we will set this to a type of **User distinguished name**, and then enter our identifier, *vpn@example.com*. You can use your own custom identifier, just follow that format. The Pre-Shared Key, also referenced by some clients as the group key, should be a reasonably strong random string, certainly much stronger than our example of *aaabbbccc*.

In order to properly ensure that the return traffic to clients works properly, make sure that Policy Generation is set to **Unique**. In order to ensure that the phase 1 negotiation goes smoothly, set Proposal Checking to **Obey**.

The Encryption Algorithm must be set to **AES** with a key length of **128 bits**. The Hash Algorithm must be set to **SHA1**. The DH key group must be set to **2 (1024 bit)**. The Lifetime must be set to *86400*.

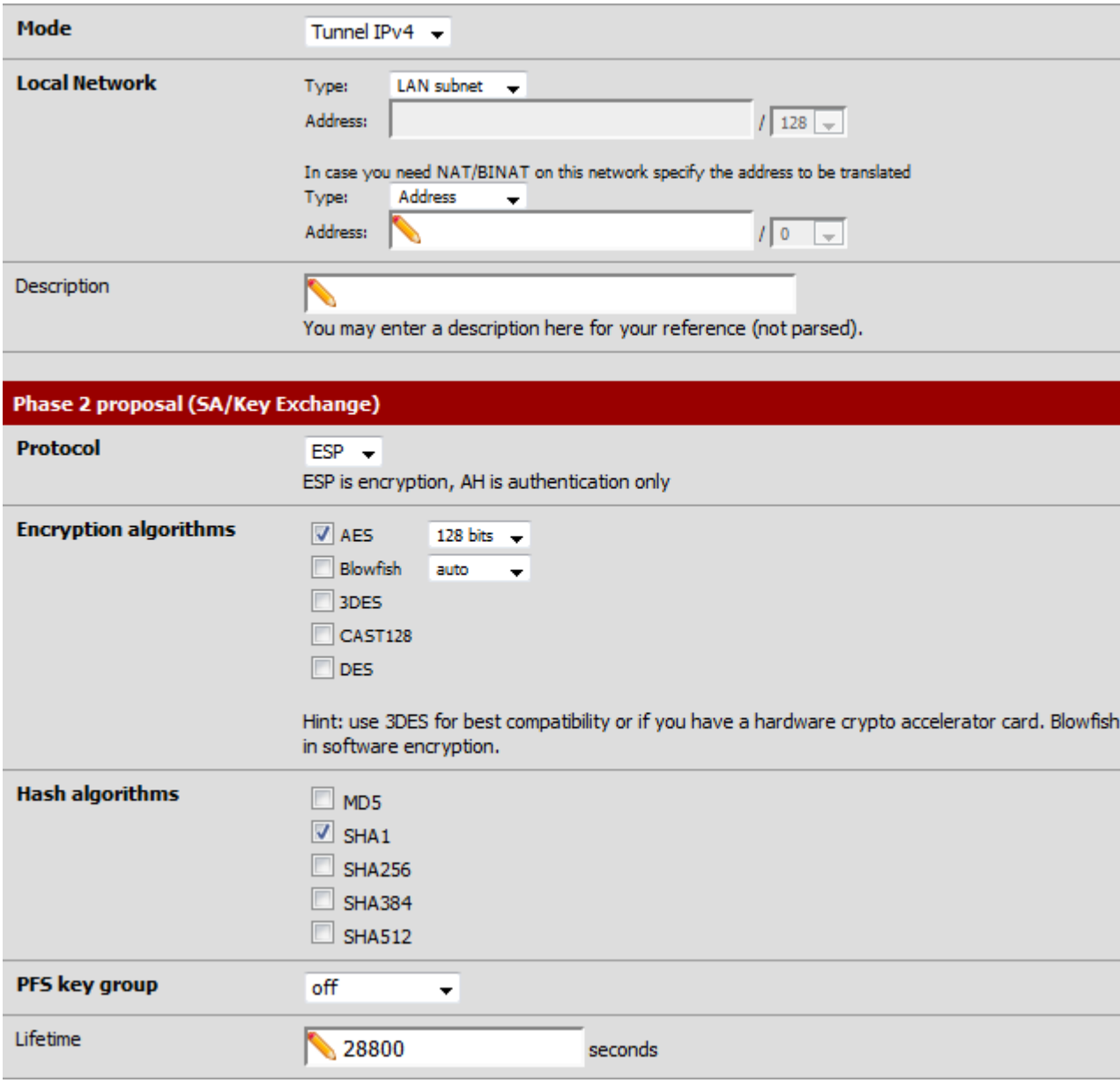
**Figure 17.22. Mobile Clients Phase 1**




You can now press Save to complete the phase 1 configuration. Next, add a phase 2 and begin that portion of the configuration.

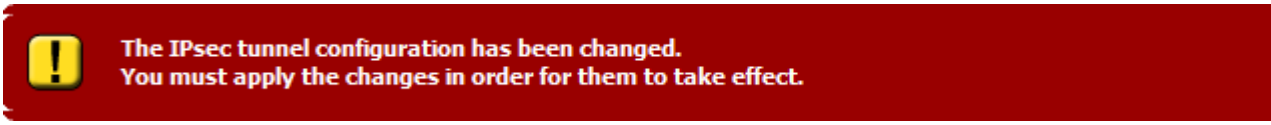
Figure 17.23, “Mobile Clients Phase 2” shows the phase 2 options for this mobile tunnel. The Mode should be set to **`Tunnel IPv4`**. The Local Network should be set to **`LAN subnet`** or another local network. The NAT settings should be left on **`None`**. The Protocol should be set for **`ESP`**, which will encrypt tunneled traffic. The Encryption algorithms for phase 2 must be set to **`AES`**, **`128 bits`** only. For Hash algorithms, you can only select **`SHA1`**. PFS must be **`off`**. The Lifetime needs to be **`28800`**. Now click Save and move on to the next step.

**Figure 17.23. Mobile Clients Phase 2**




After clicking Save, the settings must be applied before they will take effect. Click Apply changes (Figure 17.24, “Apply Mobile Tunnel Settings”) and then the tunnel setup for mobile clients is complete.

**Figure 17.24. Apply Mobile Tunnel Settings**

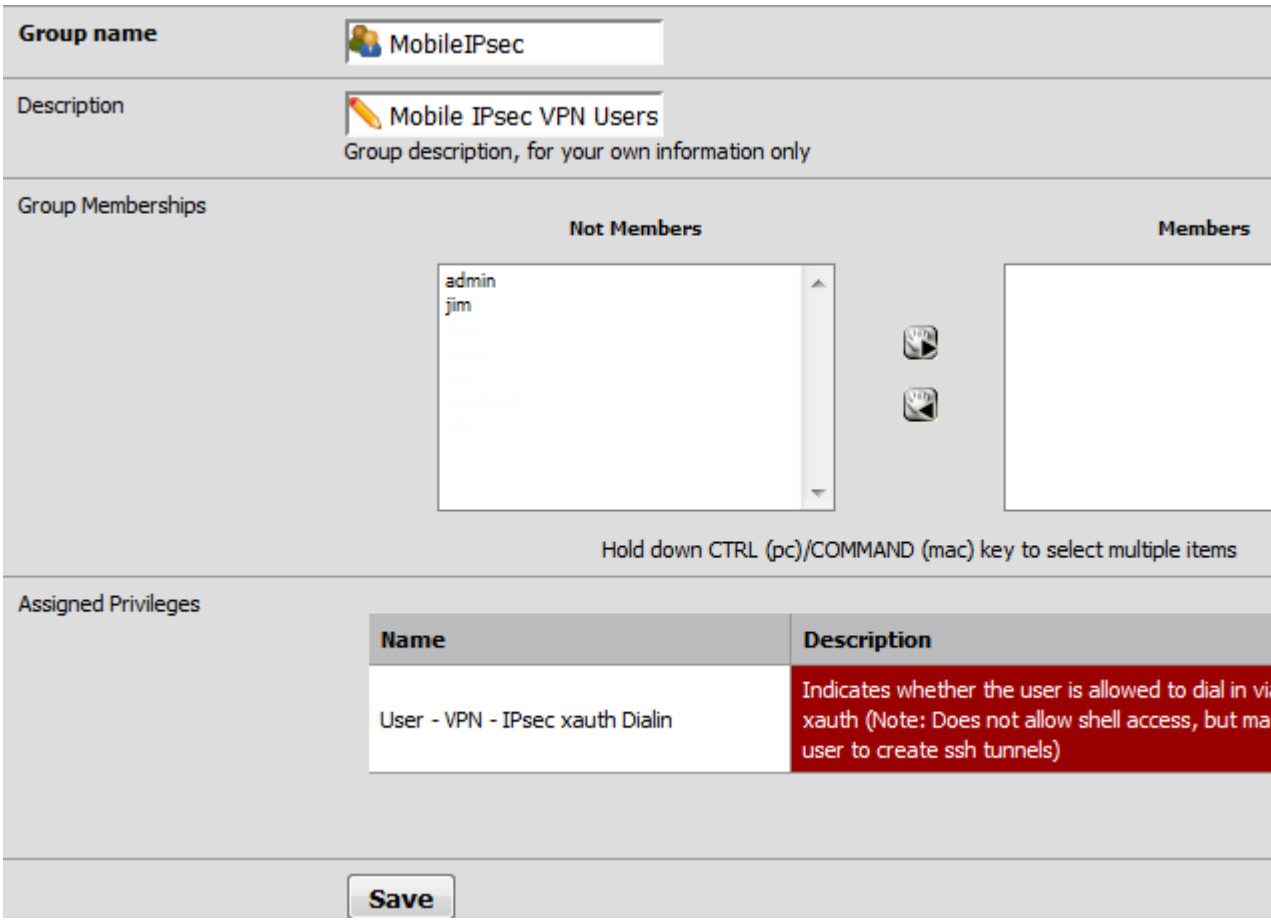



## Mobile IPsec User Creation

The next step to configuring mobile IPsec is to add users that can authentication via xauth. Since we will have multiple users, it makes sense to have an IPsec group that will have the permission needed to connect on the VPN.

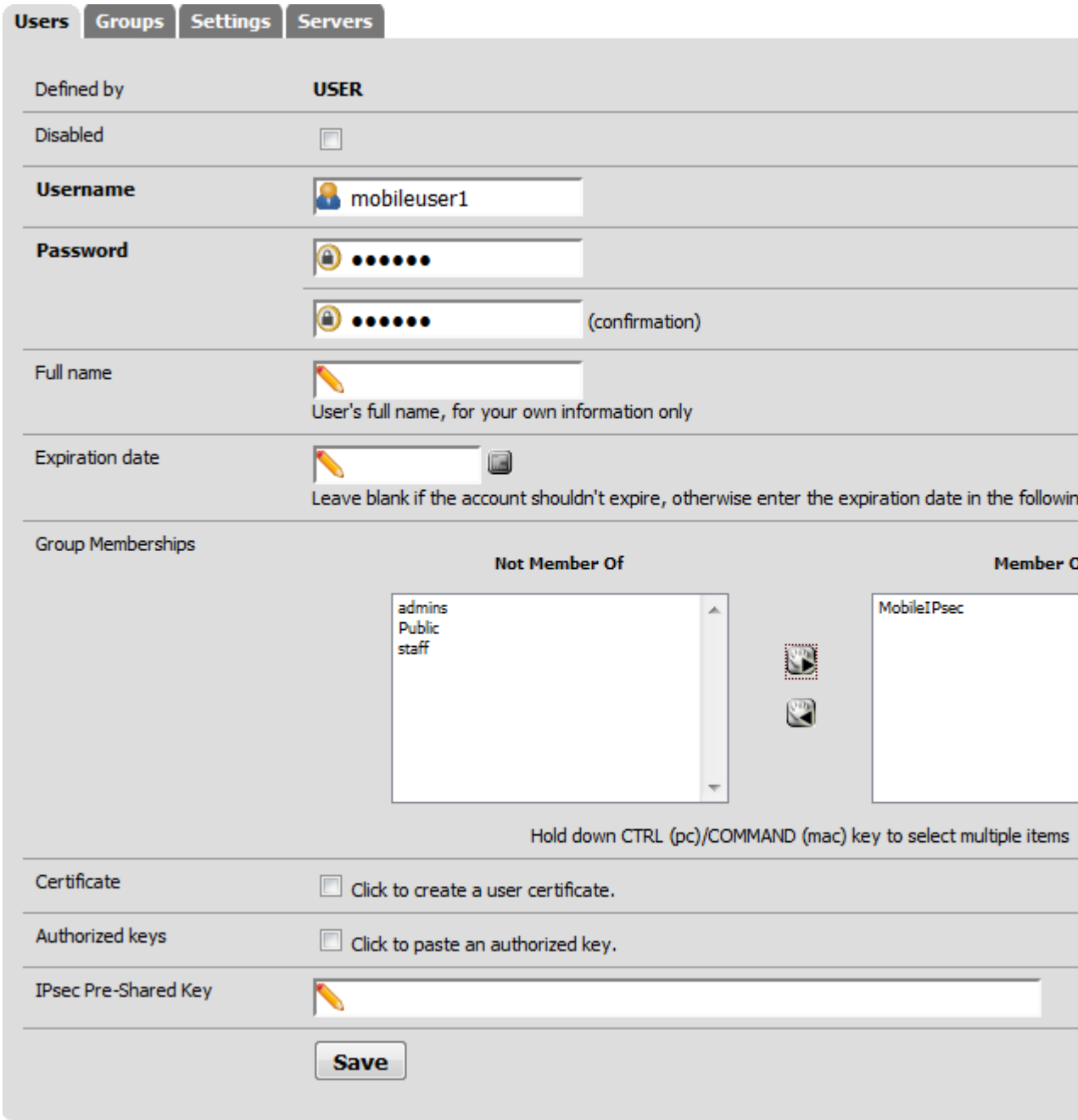
To setup the group, first go to System → User Manager on the Groups tab. Click to create a new group, set the Group Name to ***MobileIPsec***, and enter a Description such as ***Mobile IPsec VPN Users***, then press Save. Now click to edit the group so we can add its permissions. Under Assigned Privileges, click , and from the list, click once on **User - VPN - IPsec xauth Dialin**, then click Save. Now we have a group for our users. The end result of the group should look like Figure 17.25, “Mobile IPsec User Group”.

### Figure 17.25. Mobile IPsec User Group




Now we need to create users for the VPN. At System → User Manager under the Users tab, click to add a new user. Enter a Username, such as ***mobileuser1***. Enter a Password and enter it again to confirm it. In the Group Memberships list, click **MobileIPsec**, and then click the button to move it to the Member Of side. Then click Save. Repeat as many times as needed for your VPN users. A complete user is shown in Figure 17.26, “Mobile IPsec User”.

**Figure 17.26. 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 you chose for the mobile clients (or the addresses of their remote networks), and the destination will be your LAN network. For more details, the section called “IPsec and firewall rules”.

# Example Client Configuration

Each mobile client computer will need to run some kind of IPsec client software. There are many different IPsec clients available for use, some free, and some commercial applications. Typically IPsec is a fairly interoperable protocol when it comes to router-to-router tunnels, but client programs have proven more fickle, or at times incorporate proprietary extensions that are not compatible with

standards-based IPsec implementations. As mentioned before, the Cisco IPsec client included with the iPhone and iPod Touch is not compatible with pfSense IPsec, and the client provided for connecting to Watchguard Fireboxes has seen mixed results as well.

# Android Mobile IPsec

Android support for IPsec to connect to pfSense 2.x varies from version to version. Many Motorola phones such as the Droid X and Droid RAZR line have had IPsec for several versions, in the case of the Droid X, since Gingerbread (Android 2.3.x) was released. More devices gained support in Android 4.x. This section covers the two most common ways to setup an IPsec connection to the example server, depending on your device's IPsec implementation. Neither of these support getting the connected network list from the server, and thus you must define the list of networks for IPsec in the settings on the device.

Follow the set of directions that most closely matches your device.

## Note

Because Android considers using a VPN an action that should be secure, the OS will force you to use some form of locking for your device in order to protect the VPN settings. It doesn't matter which type of lock you choose (PIN lock, Pattern lock, Password, etc) but it will not let you configure a VPN until a secure lock has been added. On Android 4.x devices with Face lock, that is not available as a secure lock type.

## Motorola Android Style Mobile IPsec

From the main screen, press the Menu button, then Settings, Network & Wireless, VPN (Figure 17.27, “Motorola Android IPsec — Network Menu”), and finally Advanced IPsec VPNs (Figure 17.28, “Motorola Android IPsec — VPN Menu”).

**Figure 17.27. Motorola Android IPsec — Network Menu**

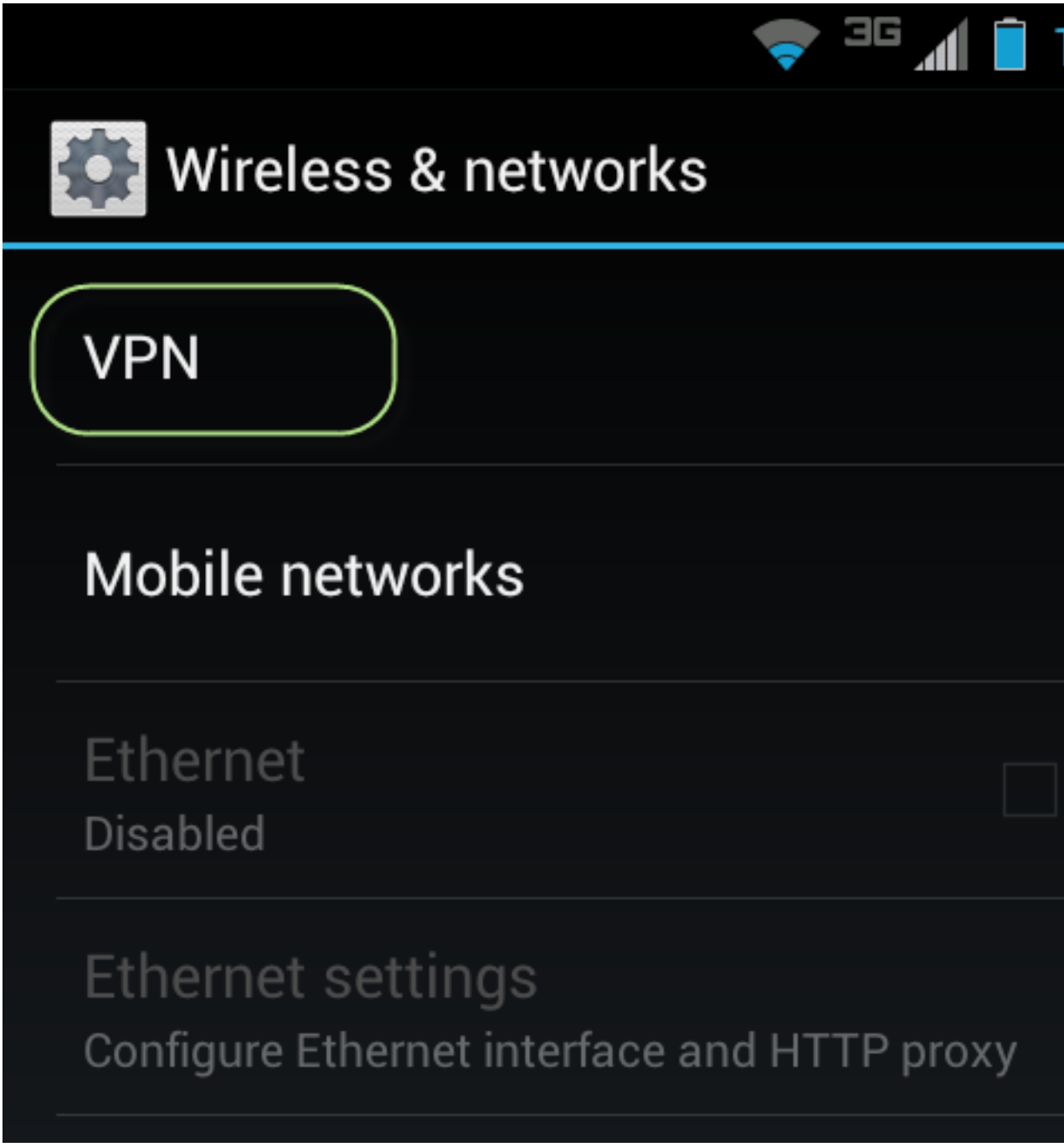
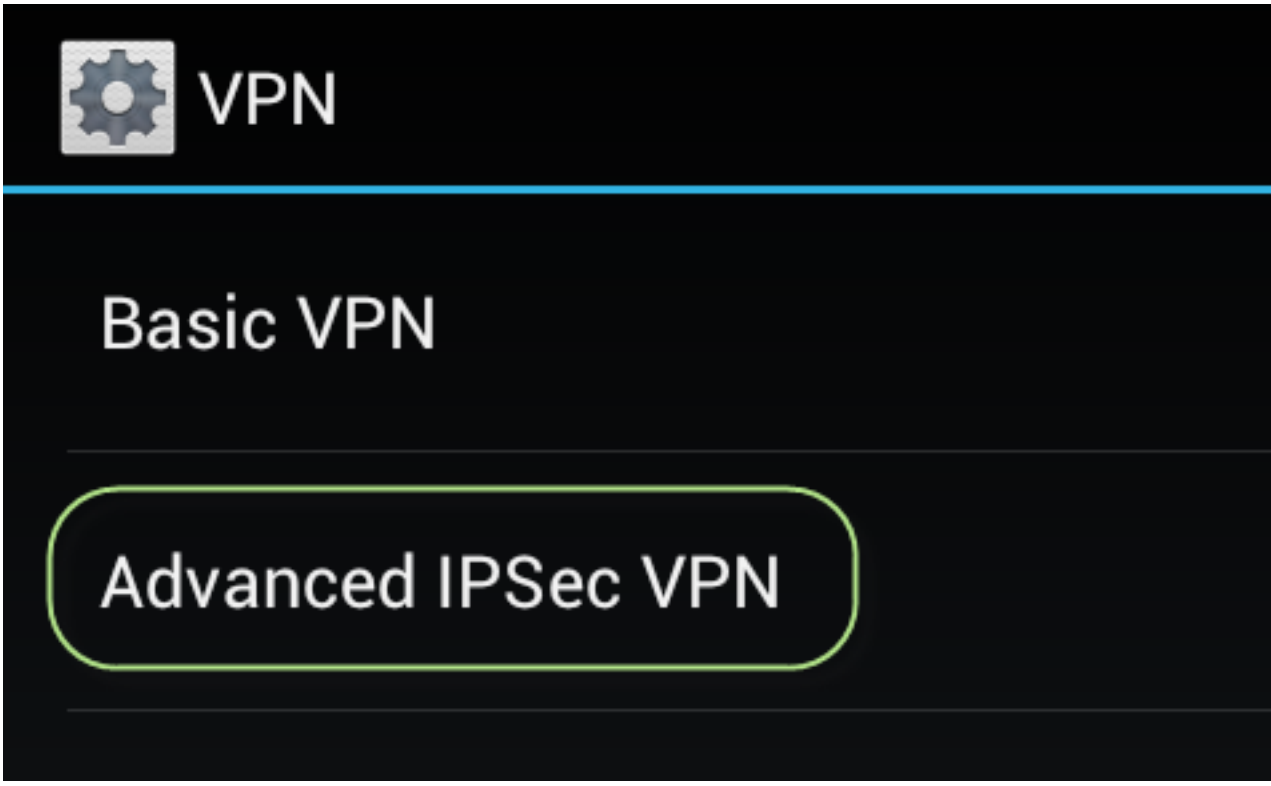
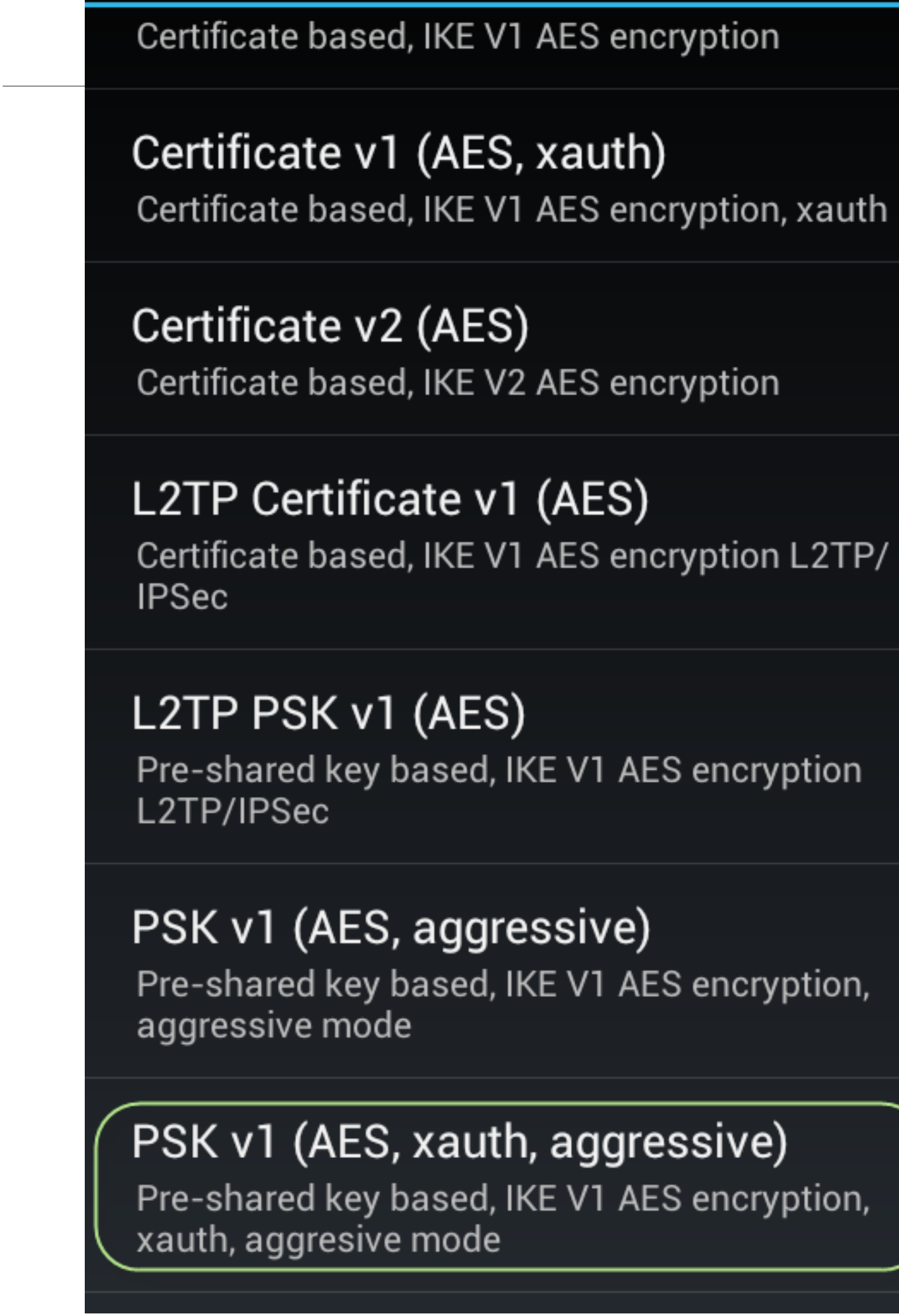


**Figure 17.28. Motorola Android IPsec — VPN Menu**




Tap Add IPsec VPN. If this choice does not appear, press the Menu button and then tap Add.

A list of IPsec VPN types is presented, as shown in Figure 17.29, “Motorola Android IPsec — IPsec Type List”, from this list, choose PSK v1 (AES, xauth, aggressive).

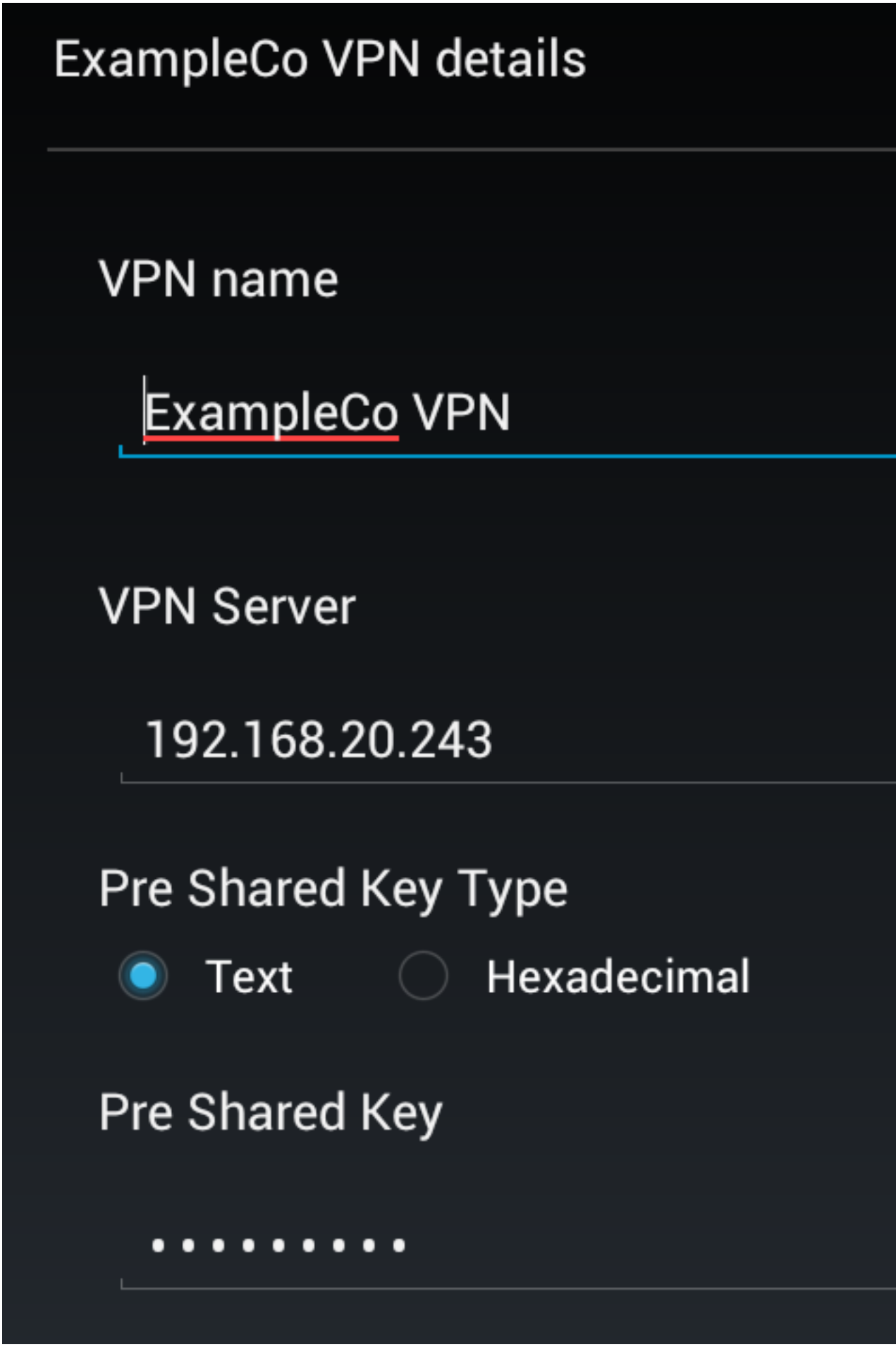

Certificate based, IKE V1 AES encryption
Certificate v1 (AES, xauth)
Certificate based, IKE V1 AES encryption, xauth
Certificate v2 (AES)
Certificate based, IKE V2 AES encryption
L2TP Certificate v1 (AES)
Certificate based, IKE V1 AES encryption L2TP/ IPSec
L2TP PSK v1 (AES)
Pre-shared key based, IKE V1 AES encryption L2TP/IPSec
PSK v1 (AES, aggressive)
Pre-shared key based, IKE V1 AES encryption, aggressive mode
PSK v1 (AES, xauth, aggressive)
Pre-shared key based, IKE V1 AES encryption, xauth, aggresive mode

For the VPN Name, enter a description of this VPN, such as *ExampleCo VPN*.

The VPN Server value should be the IP address of the server, such as *192.168.20.243*.

For Pre Shared Key Type, select **Text**, then enter the Pre Shared Key, which in this example is *aaabbbccc*.

These settings are shown in Figure 17.30, “Motorola Android IPsec — Settings”.


ExampleCo VPN details
VPN name
ExampleCo VPN
VPN Server
192.168.20.243
Pre Shared Key Type
Text
Hexadecimal
Pre Shared Key
• • • • • • • • •

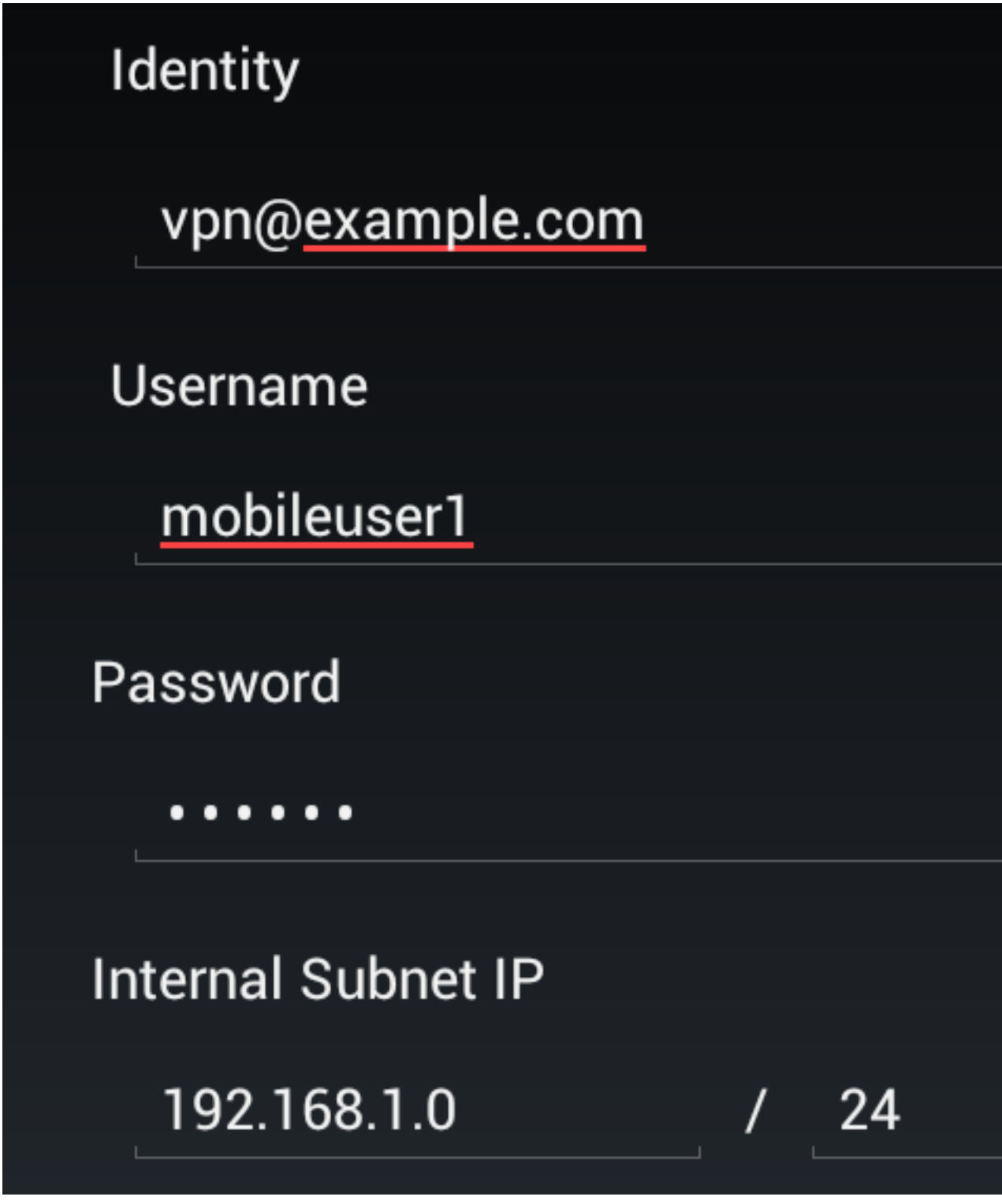
The Identity Type should be **`User FQDN`**, and then enter the Peer Identifier from the phase 1 settings. In this example, it is *`vpn@example.com`*.

Enter the Username and Password for the VPN if you wish them to be saved.

Enter the Internal Subnet IP, which is the subnet to be reached over the VPN. In this example, *`192.168.1.1`*. You may enter up to four subnets.

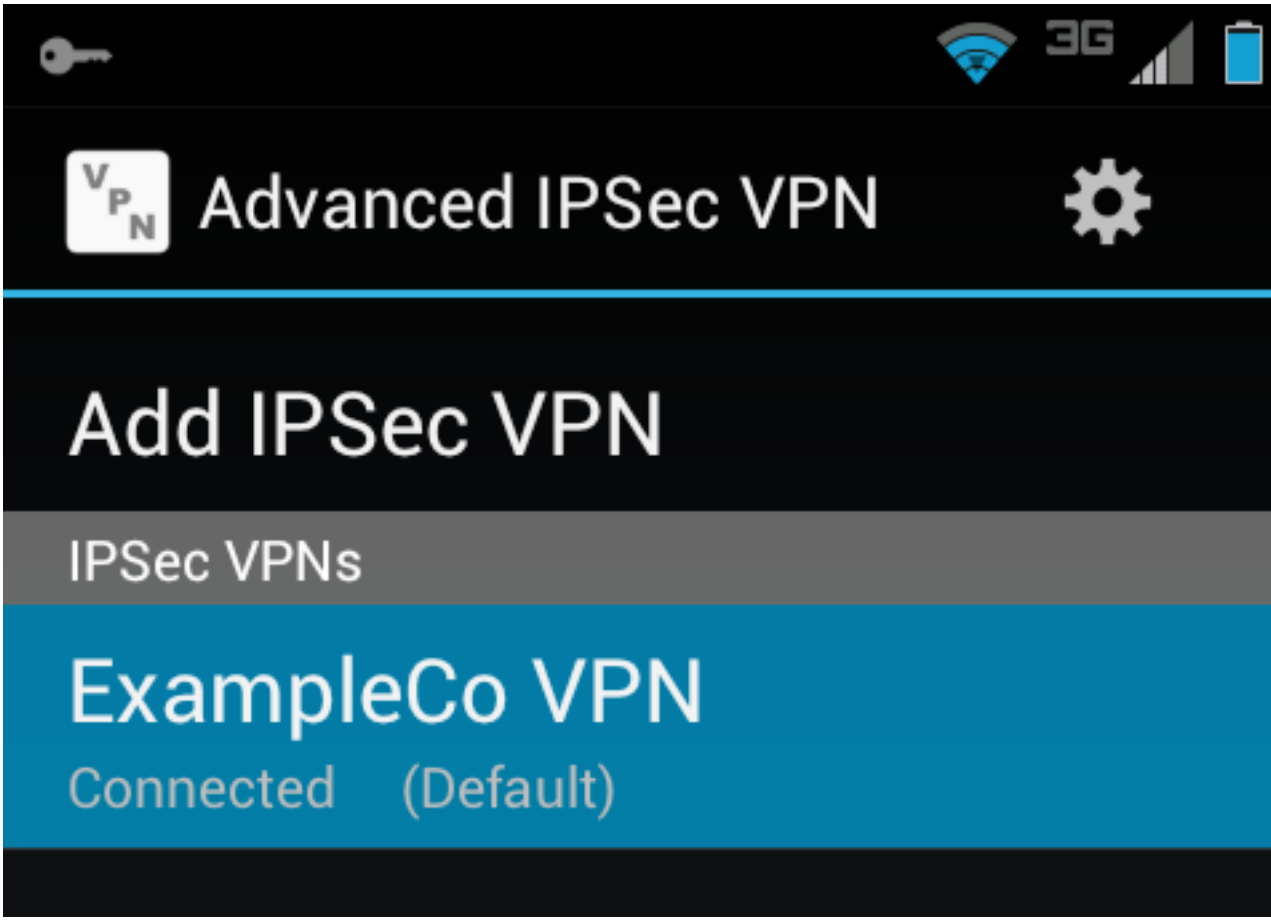
These settings are shown in Figure 17.31, “Motorola Android IPsec — Settings (continued)”.

**Figure 17.31. Motorola Android IPsec — Settings (continued)**



Tap Save, and you will be returned to the VPN list. If you tap the VPN name in the list, it will attempt a VPN connection. If it is successful, it will look like Figure 17.32, “Motorola Android IPsec — Connected”. A key icon appears in the status bar to show that the VPN is connected. You can disconnect by tapping the VPN name in the list, or by sliding down the notification bar and then tapping the VPN entry in the notification list to see the status window, which also contains a disconnect button. To edit or delete the VPN, long press on the name in the list and choose the desired action from the menu.

**Figure 17.32. Motorola Android IPsec — Connected**




## General Android 4.x Style Mobile IPsec

On Android 4.x, a different configuration method may be available if the previous style was not there.

To start, enter the system settings. This varies between phones and tablets, but would either be via the Menu button if you have one, or by tapping/sliding the notification area and then pressing the settings icon there. For example, on an Asus Transformer Prime running Jelly Bean (Android 4.1.x), you tap the clock, then the gear icon.

Under Wireless & Networks, tap More... and then Add VPN profile.

For the Name, enter a description of this VPN, such as *`ExampleCo VPN`*.

From the Type list, choose **`IPsec Xauth PSK`**, as shown in Figure 17.33, “Android 4.x IPsec — VPN Types”.

**Figure 17.33. Android 4.x IPsec — VPN Types**

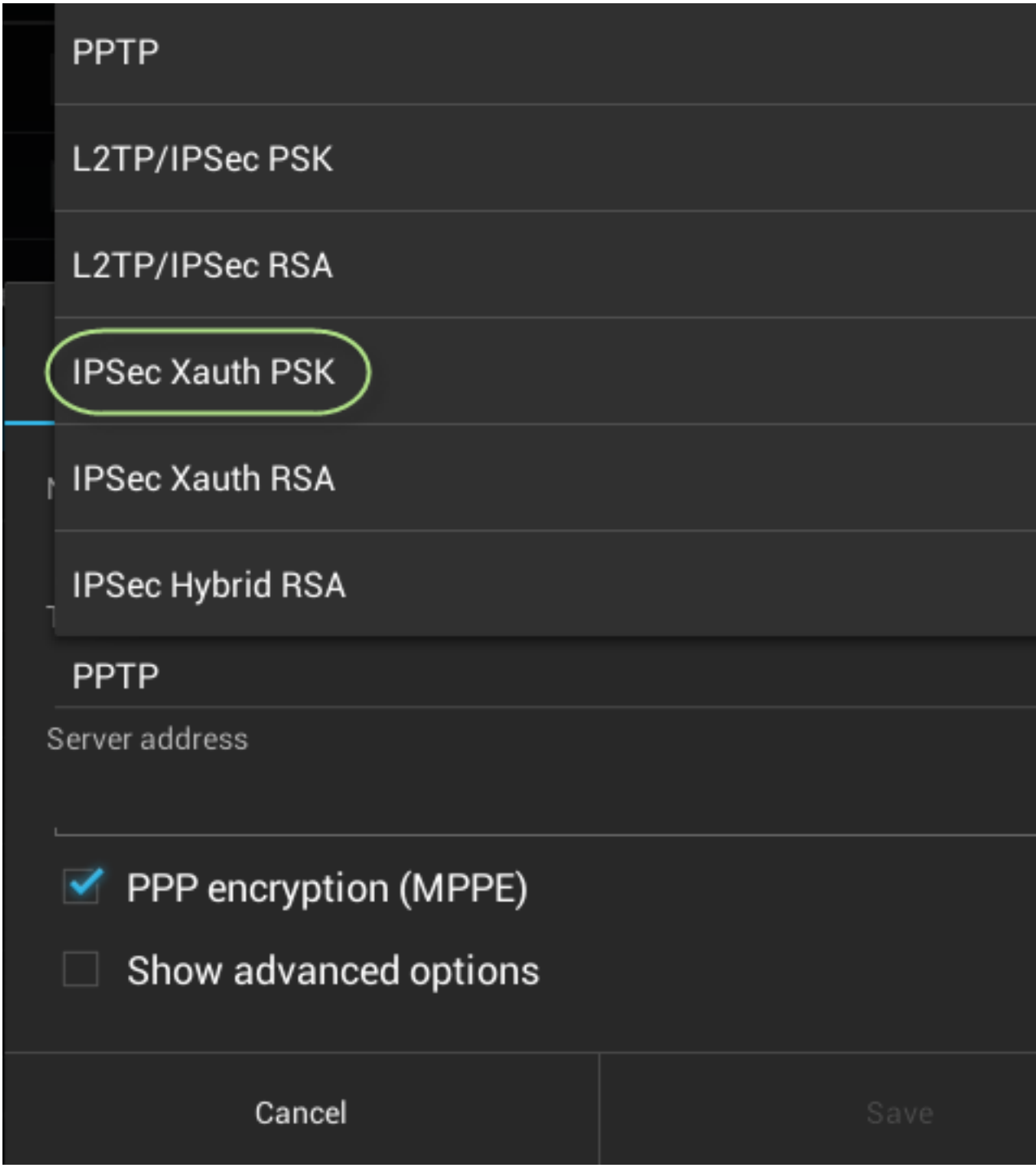
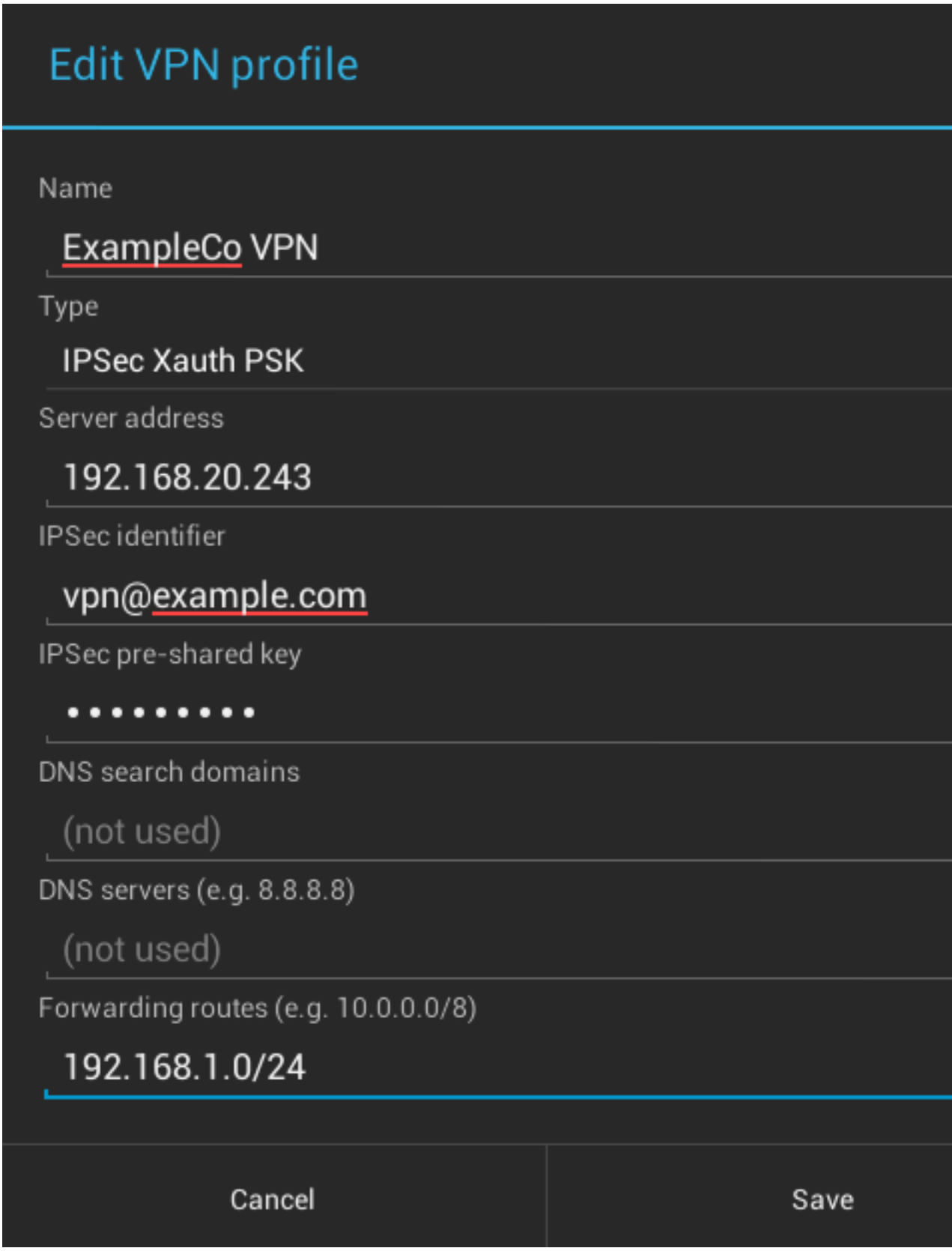


**Figure 17.34. Android 4.x IPsec — IPsec Settings**




Now enter the rest of the settings for the VPN as seen in Figure 17.34, “Android 4.x IPsec — IPsec Settings”. For the IPsec identifier, the Peer Identifier from the phase 1 settings. In this example, it is `vpn@example.com`.

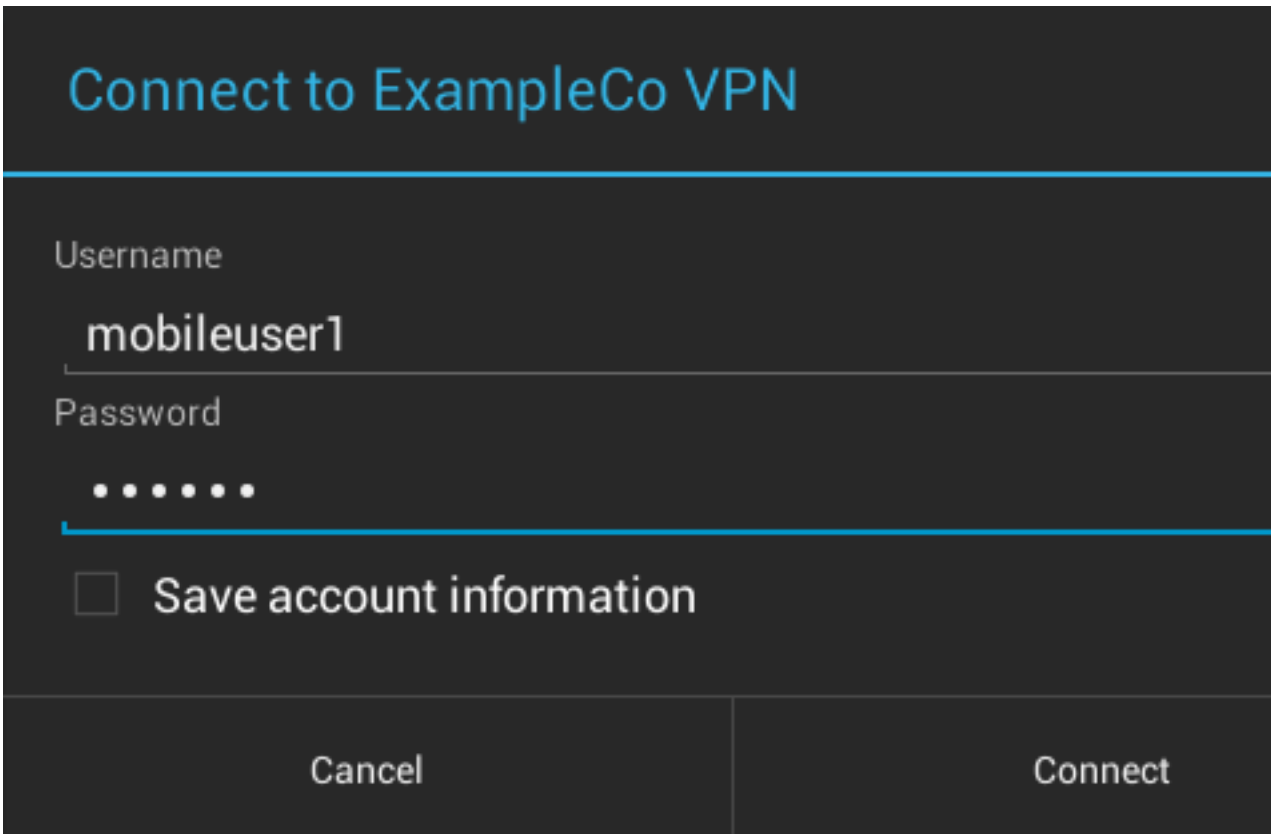
Next, enter the IPsec pre-shared key, which in this example is *aaabbbccc*.

You can leave the DNS search domains and DNS servers empty.

Under Forwarding Routes, enter the subnet to be reached over the VPN. In this example, *192.168.1.1*.

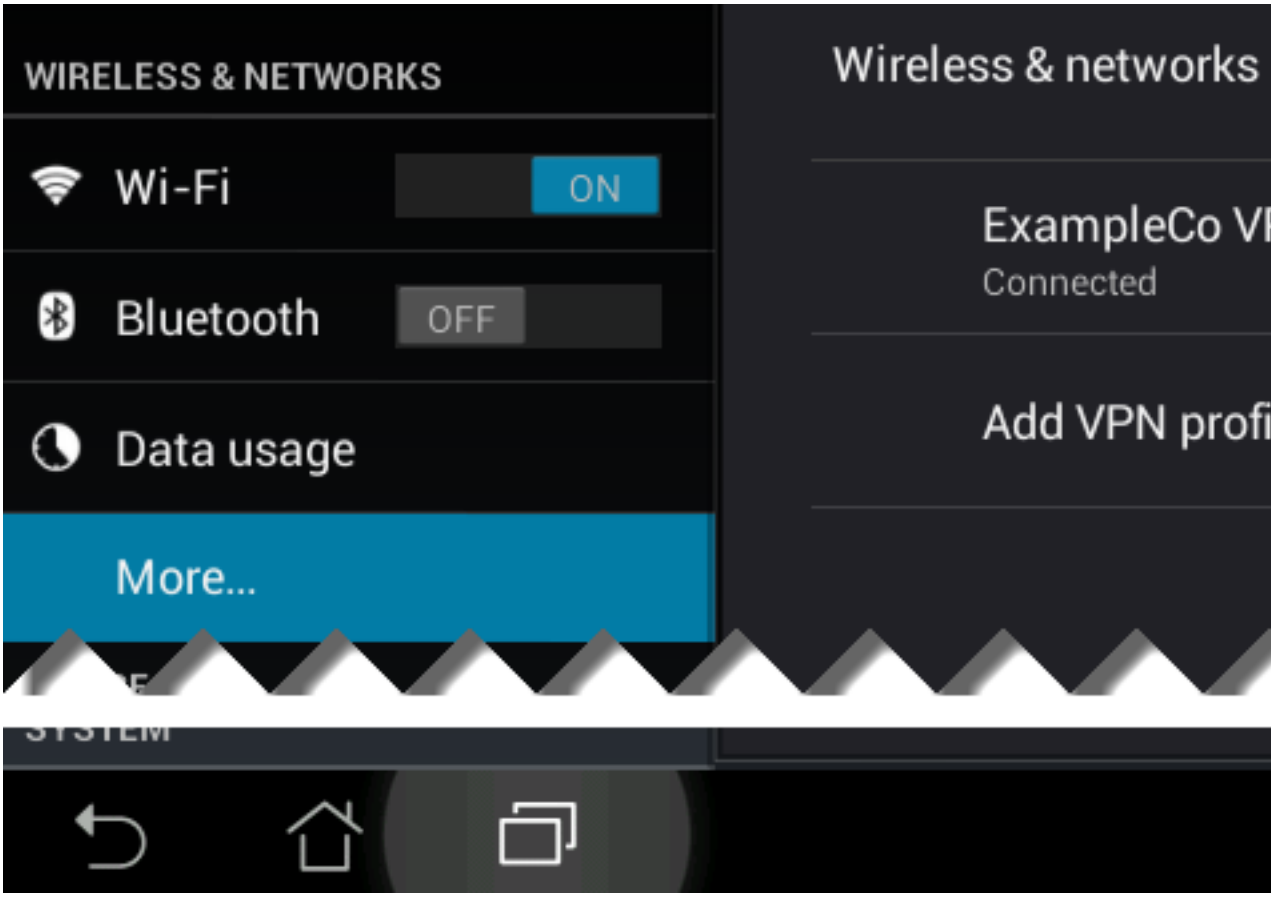
Tap Save, and you will be returned to the VPN list. If you tap the VPN name in the list, it will attempt a VPN connection. Because the credentials were not entered during the setup process, you are prompted for them now, as seen in Figure 17.35, “Android 4.x IPsec — IPsec Authentication Prompt”. Enter the Username and Password. If you want the credentials to be saved, tap Save account information.

**Figure 17.35. Android 4.x IPsec — IPsec Authentication Prompt**




If it is successful, it will look like Figure 17.36, “Android 4.x IPsec — Connected Status”. A key icon appears in the notification area to show that the VPN is connected. You can disconnect by tapping the VPN name in the list, or by sliding down the notification bar and then tapping the VPN entry in the notification list to see the status window, which also contains a disconnect button. To edit or delete the VPN, long press on the name in the list and choose the desired action from the menu.

**Figure 17.36. Android 4.x IPsec — Connected Status**




## iOS Mobile IPsec

From the home screen, tap Settings, then VPN. On older versions of iOS you may need to go from Settings, to General, then VPN. And some also require Settings, General, Network then VPN. From there, tap Add VPN Configuration....

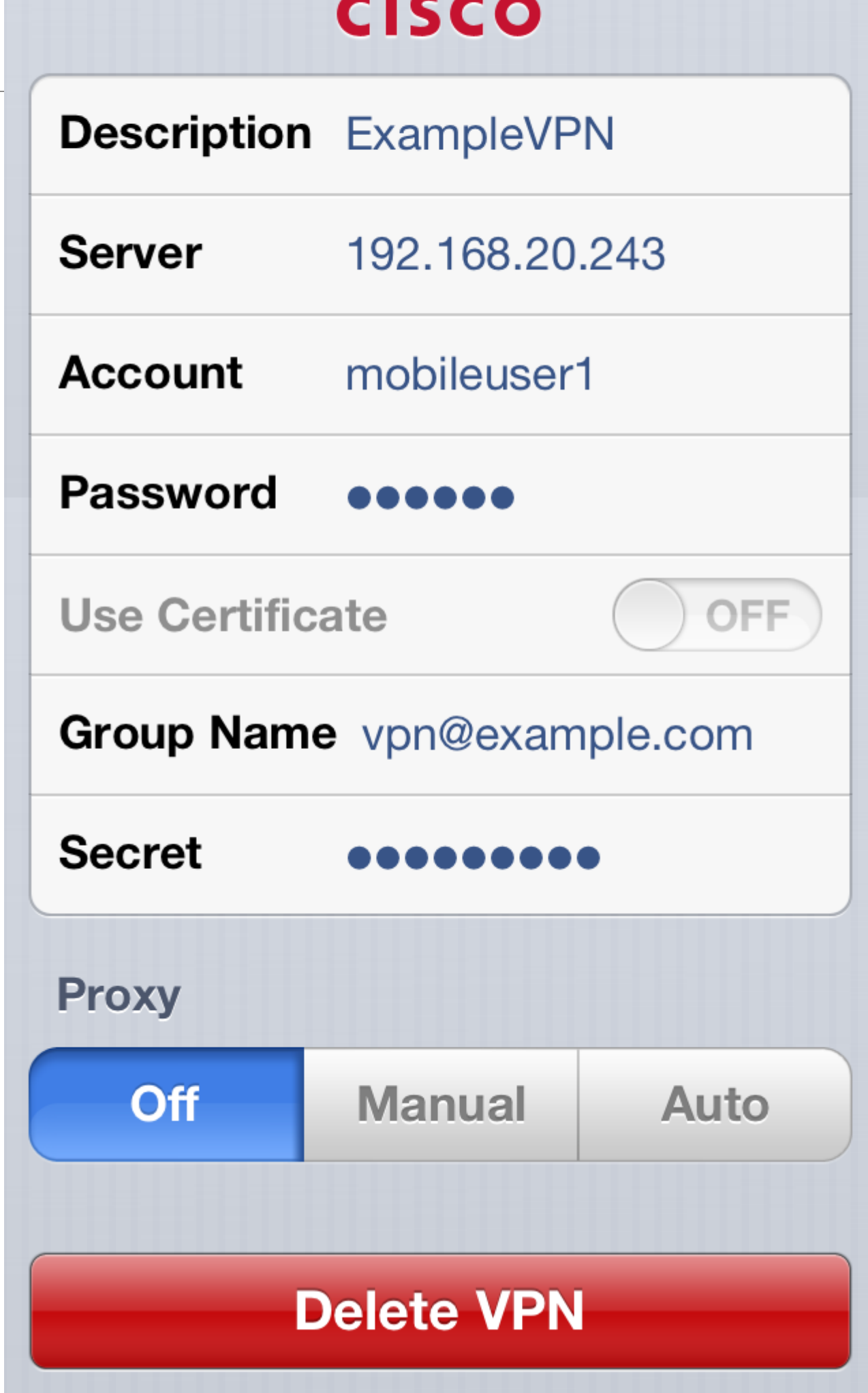
At the top of the screen, select IPsec. Enter a Description, such as *`ExampleVPN`*.

For the Server, enter the IP address or fully qualified domain name of the server. In our example, it is *`192.168.20.243`*.

Account is for this user's xauth username, such as *`mobileuser1`*. If you want the client to save the password, enter it in the Password box.

For Group Name, enter the Peer Identifier from the phase 1 settings. In this example, *`vpn@example.com`*.

The Secret field is the Pre-Shared Key from the phase 1 settings. In this example, *`aaabbbccc`*.


cisco
Description ExampleVPN
Server 192.168.20.243
Account mobileuser1
Password ●●●●●●
Use Certificate OFF
Group Name vpn@example.com
Secret ●●●●●●●●●
Proxy
Off
Manual
Auto
Delete VPN

The completed VPN setup should look like Figure 17.37, “iOS IPsec Configuration”. Press Save, then tap the VPN name to select it if you have more than one. Slide the VPN toggle to On and the VPN should connect, and the resulting status should look like Figure 17.38, “iOS IPsec — VPN Connected”. There is a VPN indicator shown in the iOS status bar to show that the VPN is active.

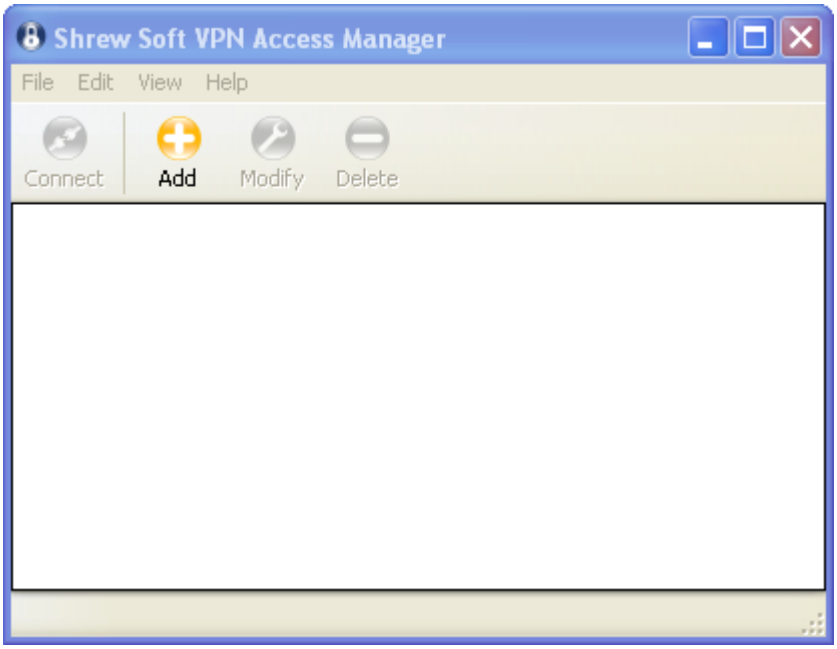
**Figure 17.38. iOS IPsec — VPN Connected**


## Shrew Soft Client for Windows

The Shrew Soft VPN Client is a solid choice for using IPsec on Windows. Not only is it easy to use and reliable, but it is also available completely free. Visit http://www.shrew.net and download the latest version of the Shrew Soft client for your platform. Run the installer, and click Next or Continue through all the prompts.

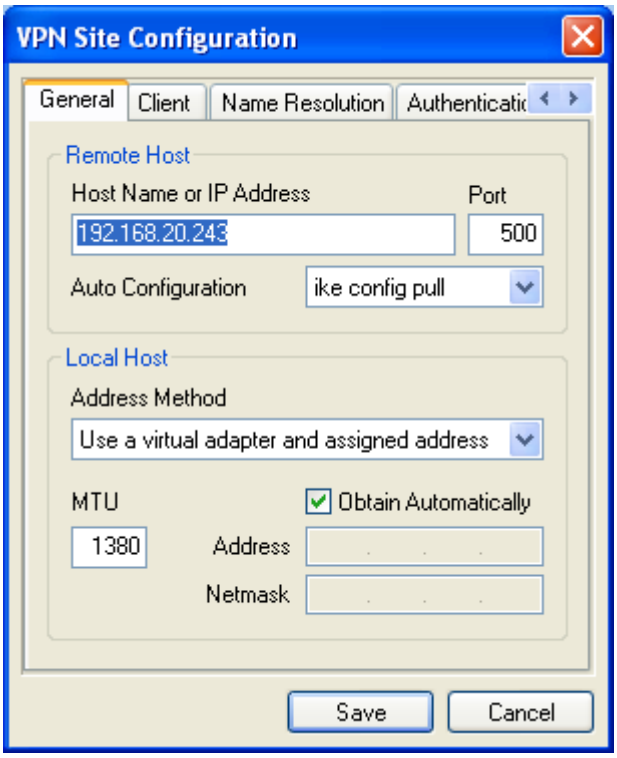
Start the Shrew Soft Client by clicking the Access Manager icon. The main screen should appear, and look like Figure 17.39, “Shrew Soft VPN Access Manager — No Connections Yet”. Next, click the Add button to start configuring a new connection.

**Figure 17.39. Shrew Soft VPN Access Manager — No Connections Yet**




The VPN Site Configuration window should open, with several tabs as in Figure 17.40, “Client Setup: General Tab”. It should start on the General tab. Here, enter the Host as the pfSense Box WAN IP, or the IP address of the pfSense interface chosen previously for IPsec use. In our example, **`192.168.20.243`**. The Port should be **`500`**. Auto Configuration should be set to **`ike config pull`** so that settings will be obtained from pfSense. For the Address Method, change that to **`Use virtual adapter and assigned address`** and then check Obtain Automatically.

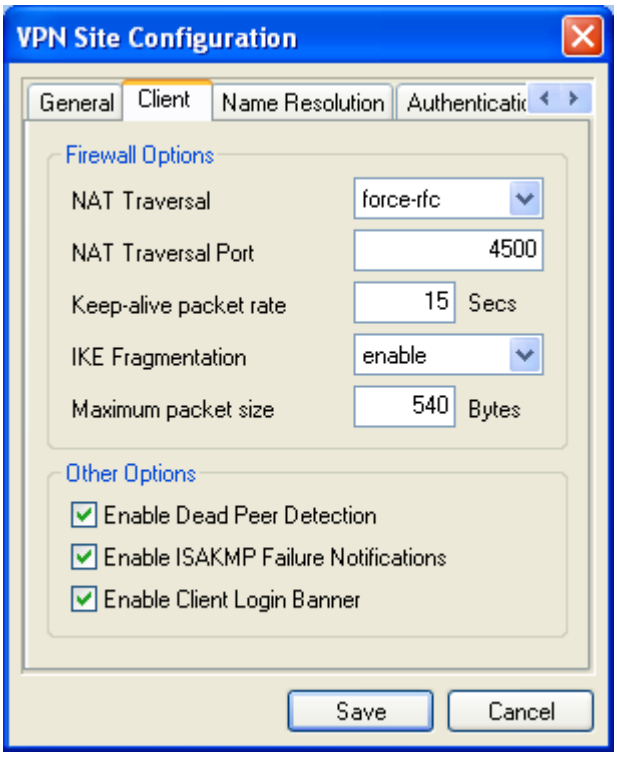
**Figure 17.40. Client Setup: General Tab**




On the Client tab, ensure that NAT Traversal is set to **`force-rfc`**, and the NAT Traversal Port should be 4500. Ensure all three checkboxes at the bottom of the screen are checked. Compare the

settings on the screen with Figure 17.41, “Client Setup: Client Tab” to be sure they match up with the proper settings.

### Figure 17.41. Client Setup: Client Tab




On the Name Resolution tab, every box should be checked and Obtain Automatically should also be checked so that the settings from the firewall will be obeyed. See Figure 17.42, “Client Setup: Name Resolution Tab” for examples.

### Figure 17.42. Client Setup: Name Resolution Tab

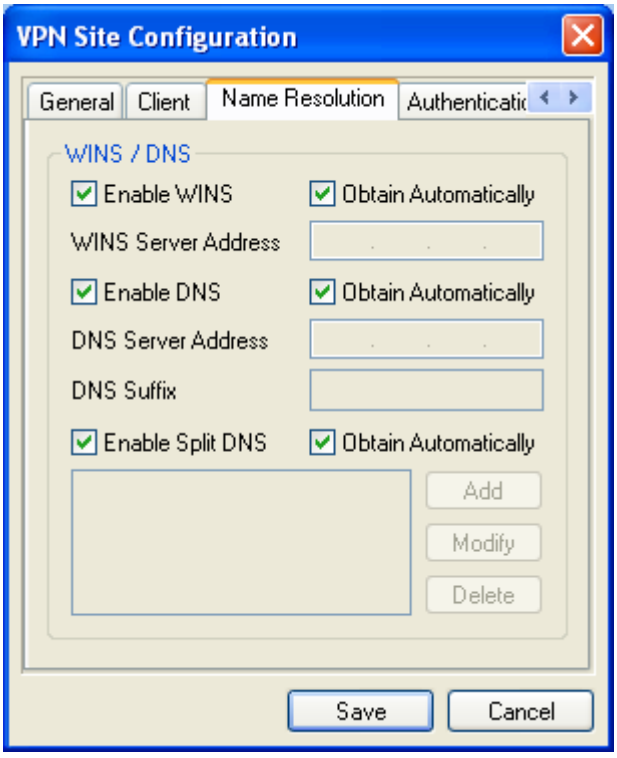
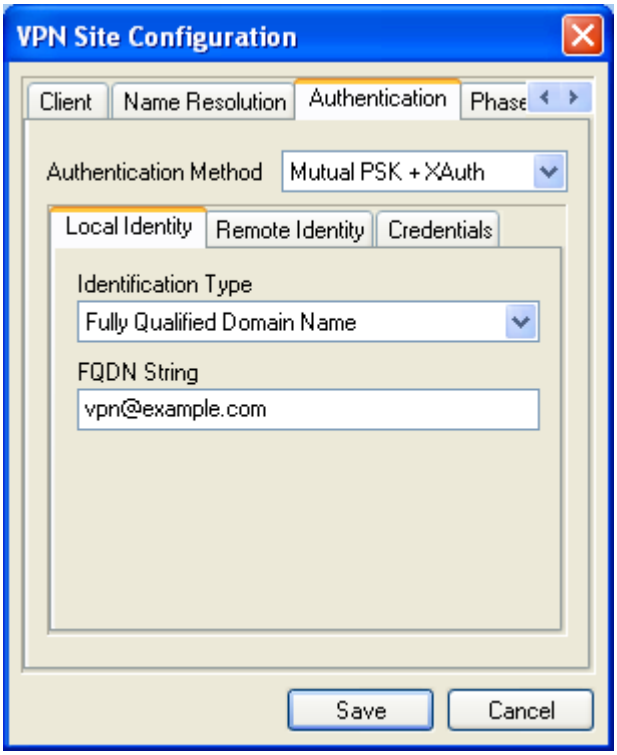


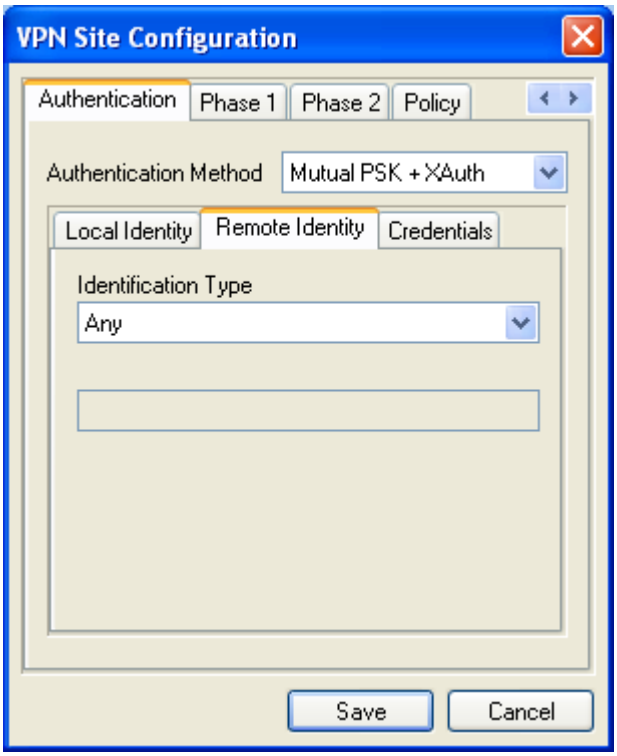
**Figure 17.43. Client Setup: Authentication, Local Identity**




The Authentication tab has three sub-tabs that need setup as well. First, set the Authentication Method to **`Mutual PSK + XAuth`** at the top, then continue on to the Local Identity tab underneath, shown in Figure 17.43, “Client Setup: Authentication, Local Identity”. Set the Identification type to **`Fully Qualified Domain Name`**, and the FQDN String to the Peer Identifier from the server's phase 1 settings, in this example, *`vpn@example.com`*.

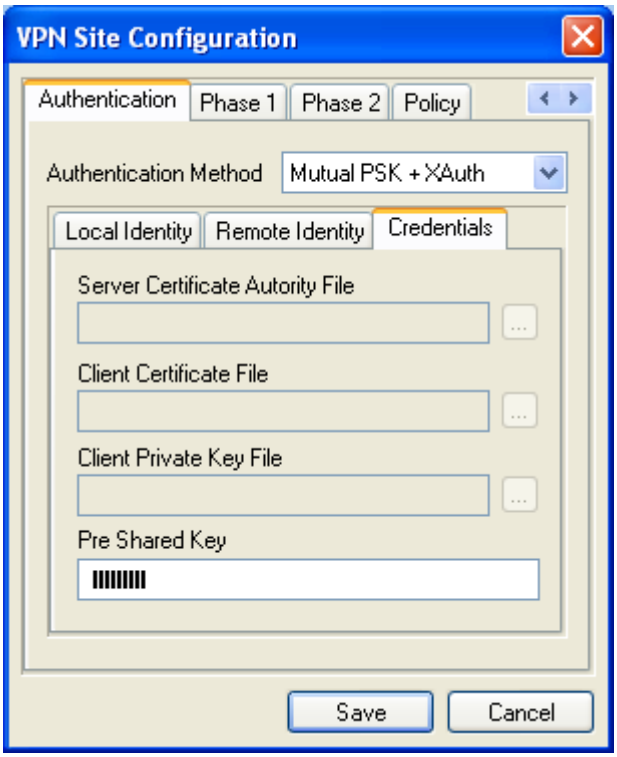
Click the Remote Identity tab (Figure 17.44, “Client Setup: Authentication, Remote Identity”). Set the Identification Type to **`Any`**.

**Figure 17.44. Client Setup: Authentication, Remote Identity**




On the Credentials tab, shown in Figure 17.45, “Client Setup: Authentication, Credentials”, fill in the Pre-Shared Key field with the key from the phase 1 settings, in our example, *aaabbbccc*.

**Figure 17.45. Client Setup: Authentication, Credentials**




Now go back up to the Phase 1 tab, seen in Figure 17.46, “Client Setup: Phase 1”. These settings will match up with those set on the server tunnel's Phase 1 section. Set the Exchange Type to

**aggressive**, the DH Exchange to **Group 2**, Cipher Algorithm to **AES**, Cipher Key Length to **128** Bits, Hash Algorithm to **SHA1**, and Key Life Time limit to **86400**.

**Figure 17.46. Client Setup: Phase 1**

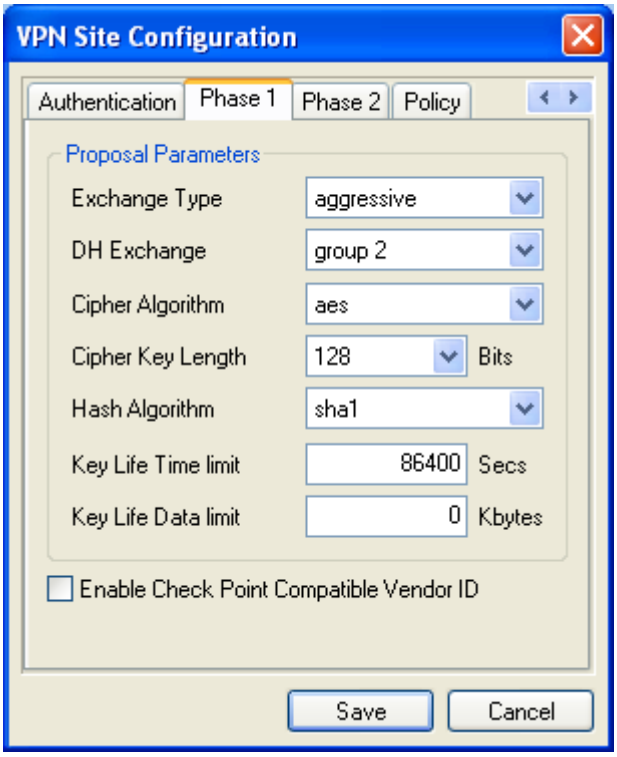
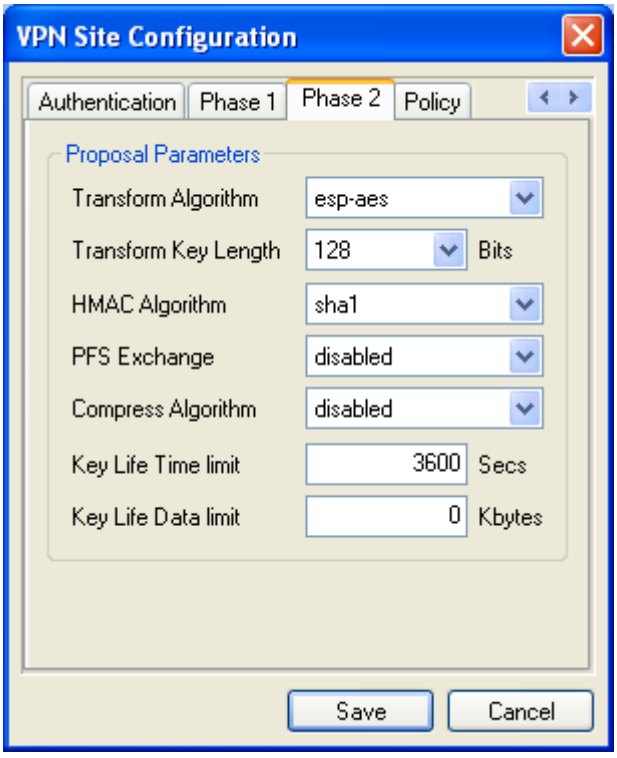



**Figure 17.47. Client Setup: Phase 2**

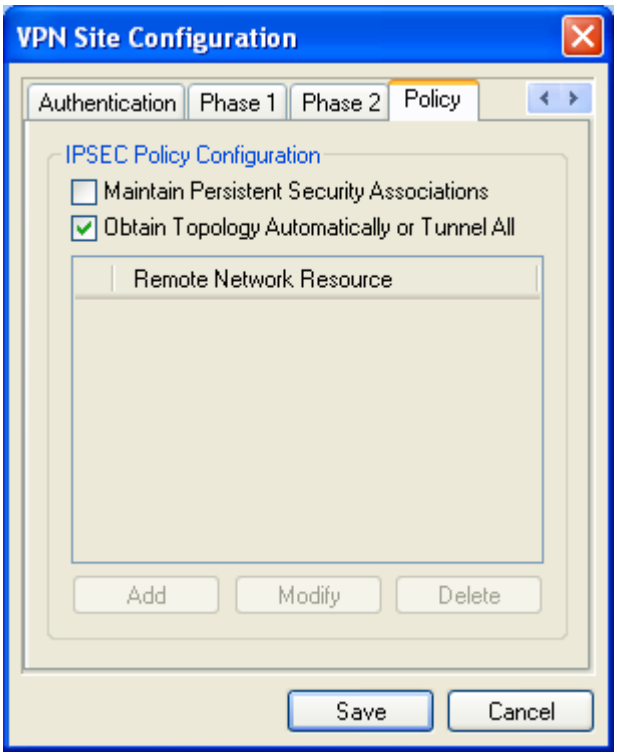



The settings on the Phase 2 tab will also be the same as those set on the mobile clients Phase 2 section, as can be seen in Figure 17.47, “Client Setup: Phase 2”. Set the Transform Algorithm to **esp-**

**aes**, Transform Key Length to **128** Bits, HMAC Algorithm is **SHA1**, PFS is **disabled**, Compress Algorithm is **disabled**, and Key Life Time limit is **3600**.

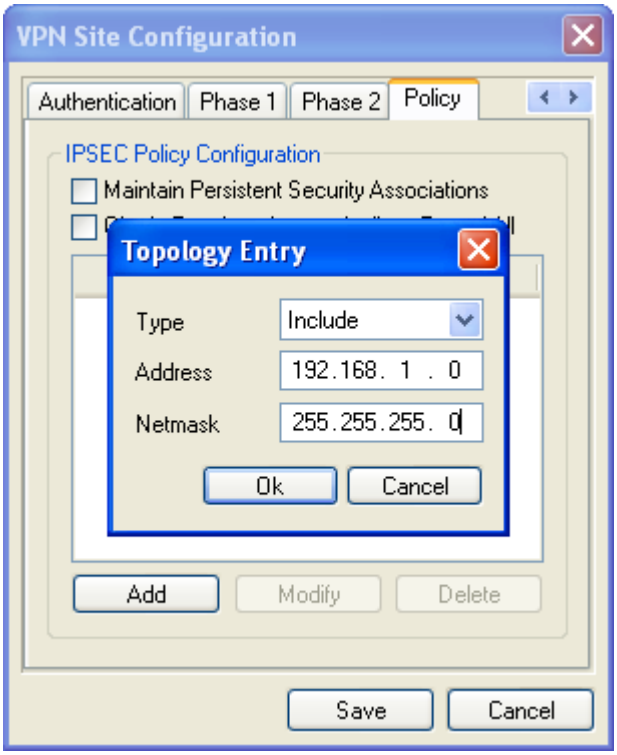
Finally there is the Policy tab, shown in Figure 17.48, “Client Setup: Policy”. This controls what traffic will be sent on the tunnel. Uncheck Obtain Topology Automatically, then click the Add button.

### Figure 17.48. Client Setup: Policy



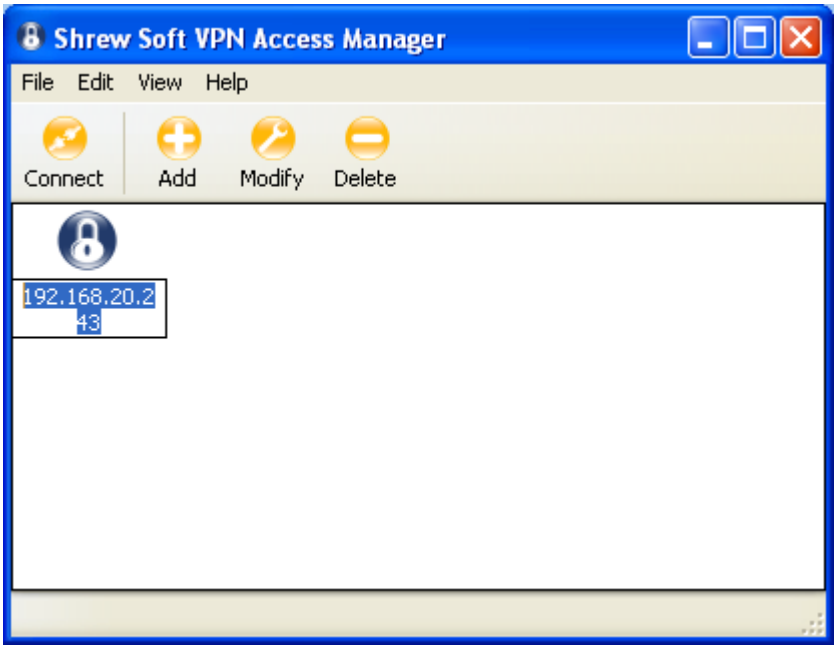

On the Topology Entry screen, seen in Figure 17.49, “Client Setup: Policy, Add Topology”, check Obtain Topology Automatically or Tunnel All. This will make the client pull the list of networks from the server, or tunnel all traffic if the server does not specify any. Alternately, you can leave that option unchecked and specify what subnet will be on the other end of the tunnel. To do that, click Add, then set the Type to **Include**. For the Address, enter the network behind pfSense on the other side, and the Netmask that goes along with it. For our example that will be **192.168.1.0** and **255.255.255.0** respectively. Click OK.

**Figure 17.49. Client Setup: Policy, Add Topology**



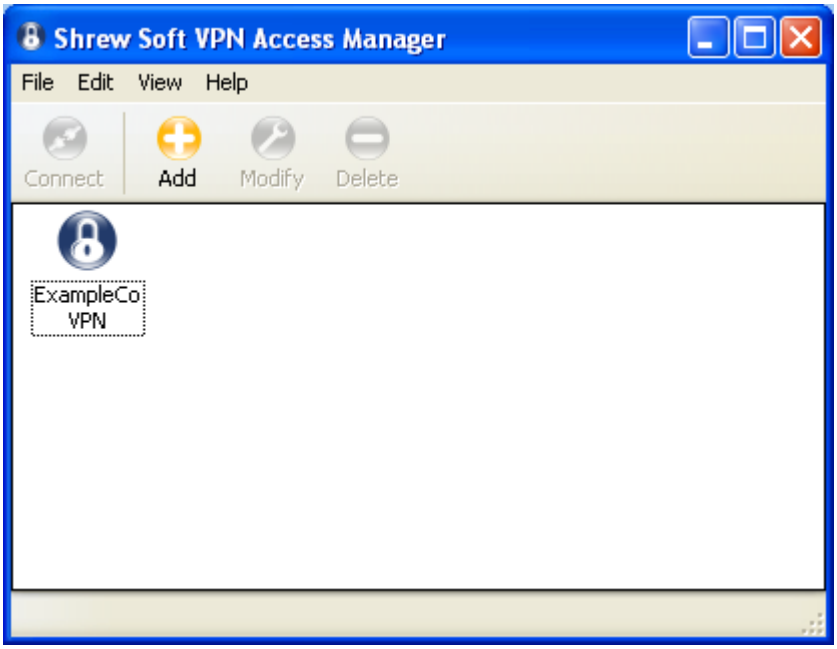

When you click Save, you will be taken back to the main screen of the Shrew Soft client, and you have an opportunity to change the name of the connection, as in Figure 17.50, “Client Setup: New Connection Name”.

**Figure 17.50. Client Setup: New Connection Name**




It is a good idea to name the connection after the location to which it connects. In this case, I named it after the office where the tunnel leads as Figure 17.51, “Ready To Use Connection” shows.

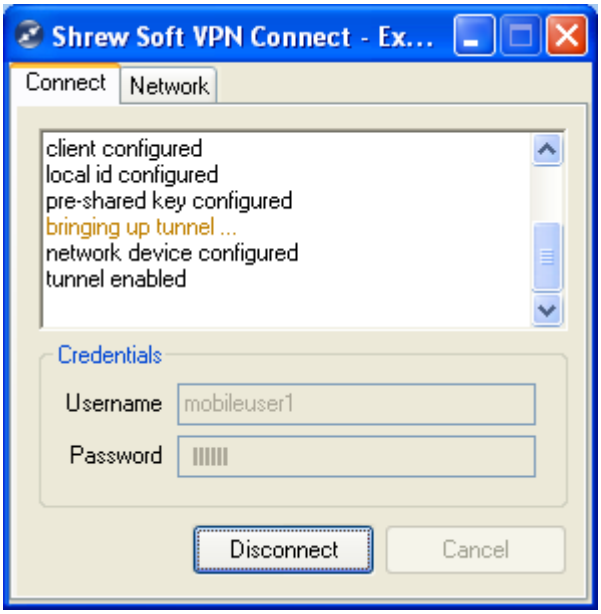
**Figure 17.51. Ready To Use Connection**




To connect to that VPN, click it once to select it and then click Connect. The VPN connect dialog will appear. Now, enter the Username and Password, and then click the Connect button on there. If the tunnel is successfully established, it will be indicated in the window. Figure 17.52, “Tunnel Authentication Prompt” shows the output from a successful connection.

**Figure 17.52. Tunnel Authentication Prompt**

**Figure 17.53. Connected Tunnel**




You should now be able to contact systems at the other end of the tunnel. If it didn't come up right or pass traffic, double check all of the settings on both sides as they are listed here. Otherwise, continue on to the troubleshooting section.

## TheGreenBow IPsec Client

TheGreenBow IPsec Client is a commercial VPN client for Windows which is compatible with pfSense. Instructions for configuring this client with pfSense can be found in the VPN gateway support [http://www.thegreenbow.com/vpn_gateway.html] section of their website. For more information about purchasing and configuring the client, visit their website [http://www.thegreenbow.com]. They offer a free 30 day trial of the client for those looking to evaluate it as a possible solution.

## NCP Secure Entry Client

The Secure Entry Client by NCP [http://www.ncp-e.com/en/downloads/software.html] is another commercial IPsec client for Windows, Windows Mobile, and Symbian. As it is standards-compliant, it can also connect to pfSense systems.

## SSH Sentinel

SSH Sentinel is another standards-compliant IPsec client for Windows. Though SSH Sentinel does work with pfSense, its configuration is quite complex and the available free client is ten years old, having been released in 2002. Due to those factors, we do not recommend its use, and the Shrew Soft client should be used in its place.

## IPSecuritas

IPSecuritas by Lobotomo Software [http://www.lobotomo.com/products/IPSecuritas/] is a freeware Mac OS X client for IPsec that some users have reported to work with pfSense.

## Linux Clients

There are some freely available Linux clients, but they vary between distributions. Some are just front-ends to other utilities like **ipsec-tools**, but they should work as long as the client configurations are similar to the one demonstrated previously.

## Cisco VPN Client

The Cisco VPN Client does not currently work with pfSense because of the way it handles tunneled traffic. This should work now that xauth is possible, but so far we haven't seen any reports of long-term success.

# Testing IPsec Connectivity

The easiest test for an IPsec tunnel is a ping from one client station behind the router to another on the opposite side. If that works, the tunnel is up and working properly.

As mentioned in the section called “pfSense-initiated Traffic and IPsec”, traffic initiated from pfSense will not normally traverse the tunnel without some extra routing, but there is a quick way to test the connection from the router console using the **ping** command while specifying a source address with the `-S` parameter. Without using `-S` or a static route, the packets generated by **ping** will not attempt to traverse the tunnel. This would be the syntax to use 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 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 routers. Given our site-to-site example above, this is what you would type to test from the console of the Site A router:

```
# ping -S 192.168.1.1 10.0.10.1
```

You should receive ping replies from Site B's LAN address if the tunnel is up and working properly. If you do not receive replies, move on to the troubleshooting section (the section called “IPsec Troubleshooting”).

# IPsec Troubleshooting

Due to IPsec's finicky nature, 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.

## Tunnel does not establish

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 (especially with mobile clients) that a router involved on one side or the other does not properly handle IPsec traffic, primarily where NAT is involved. The problems are generally with the ESP protocol. NAT Traversal (NAT-T) encapsulates ESP in UDP port 4500 traffic to get around these issues, but is not currently available in pfSense.

In the case of a timeout on a mobile client, first check the service status at Status → Services. If the service is stopped, double check that Allow mobile clients is checked on VPN → IPsec, Mobile clients tab. 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.

# Tunnel establishes but no traffic passes

The top suspect if a tunnel comes up but won't pass traffic would be the IPsec firewall rules. If you are at Site A and cannot reach Site B, check the Site B router. Conversely, if you are at Site B and cannot contact Site A, check Site A. Before looking at the rules, be sure to check the firewall logs which are at Status → System Logs, on the Firewall tab. If you see blocked entries involving 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 the section called “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 rules on the IPsec interface. Blocked packets on the LAN or other internal interface may indicate that an additional rule may be needed on that interface's 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 should allow the tunnel to establish. In the case of mobile tunnels, you will need to 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 Chapter 10, *Firewall* for more information on how to properly create and troubleshoot firewall rules.

In some cases it may also be possible that a setting mismatch could also cause traffic to fail passing the tunnel. In one instance, I saw a subnet defined on one non-pfSense router as 192.168.1.1/24, and on the pfSense side it was 192.168.1.0/24. The tunnel established, but traffic would not pass until the subnet was corrected.

There could also be an issue with how the packets are being routed. Running a **traceroute** (**tracert** on Windows) to an IP on the opposite side of the tunnel may be enlightening. Repeat the test from both sides of the tunnel. Check the the section called “Routing and gateway considerations” section in this chapter for more information. When using **traceroute**, you will see that traffic which does enter and leave 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 the section called “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.

1. 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 (see the section called “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 you can do about that other than getting the software on the device fixed. You can verify this by running **tcpdump** on the inside interface of the firewall connected to the network containing the device. Troubleshooting with **tcpdump** is covered in the section called “Using tcpdump from the command line”, and an IPsec-specific example can be found in the section called “IPsec tunnel will not connect”. If you see traffic going out the inside interface on the firewall, but no replies coming back, the device is not properly routing its reply traffic (or could potentially be blocking it via a firewall).

2. 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.

3. Host firewall — if there is a firewall on the target host, it may not be allowing the connections.

4. Firewall rules on pfSense — ensure the rules on both ends allow the desired network traffic.

# Connection Hangs

Historically, IPsec has not gracefully handled 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.), you may need to setup MSS clamping for the VPN. That can be activated under System → Advanced on the Miscellaneous 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 you find the breaking point, then back off a little from there. If that does not help, the WAN MTU may need reduced. A reduced MTU will ensure that the packets traversing the tunnel are all of a size which can be transmitted whole, similar to MSS clamping but for all traffic and not just for TCP. Similar value advice applies to the MTU as applies to MSS, start at 1300 and work your way up to higher values.

# "Random" Tunnel Disconnects/DPD Failures on Embedded Routers

If you experience dropped IPsec tunnels on an ALIX or other embedded hardware, you may need to disable DPD on the tunnel. You may be able to correlate the failures to 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.

# Tunnels Establish and Work but Fail to Renegotiate

In some cases, you may find that 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.

## NAT Traversal Causing Renegotiation Failure

If both sides of the tunnel have public IPs, and NAT-T is enabled on one or both sides, we have seen situations where this has led to a problem with renegotiation. Disable NAT-T on both sides and the tunnel should reestablish properly.

## 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.

You can also try to uncheck Prefer Old IPsec SA under System → Advanced on the Miscellaneous tab.

# Tunnel Establishes When Initiating, but not When Responding

If the tunnel establishes properly when pfSense initiates the tunnel, but not when the other end initiates, this usually indicates there is a slight mismatch in the Phase 1 settings that was ignored because a more secure value was used by the peer. The easiest way around this is to set the Proposal Checking option on the tunnel's phase 1 settings to **Obey**.

# IPsec Log Interpretation

The IPsec logs available at Status → System Logs, on the IPsec tab will contain a record of the tunnel connection process. In this section, we will demonstrate some typical log entries, both good and bad. The main things to look for are key phrases that indicate what part of a connection actually worked. If you see "ISAKMP-SA established", that means phase 1 was completed successfully and a Security Association was negotiated. If "IPsec-SA established" is seen, then phase 2 has also been completed and the tunnel should be up and working at that point.

In the following examples, the tunnel is being initiated from Site A.

## Successful Connections

These are examples of successful tunnels, in both Main Mode and Aggressive.

### Successful Main Mode Tunnel

Log output from Site A:

```
ERROR: such policy already exists. anyway replace it: 192.168.30.0/24[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.30.1/32[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.30.0/24[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.32.0/24[0] 192.168
[ToSiteB]: INFO: IPsec-SA request for 172.16.3.41 queued due to no phase1 found
[ToSiteB]: INFO: initiate new phase 1 negotiation: 172.16.0.40[500]<=>172.16.3.4
INFO: begin Identity Protection mode.
INFO: received Vendor ID: DPD
INFO: received broken Microsoft ID: FRAGMENTATION
[ToSiteB]: INFO: ISAKMP-SA established 172.16.0.40[500]-172.16.3.41[500] spi:cc3
[ToSiteB]: INFO: initiate new phase 2 negotiation: 172.16.0.40[500]<=>172.16.3.4
[ToSiteB]: INFO: IPsec-SA established: ESP 172.16.3.41[0]->172.16.0.40[0] spi=94
[ToSiteB]: INFO: IPsec-SA established: ESP 172.16.0.40[500]->172.16.3.41[500] sp
```

Log output from Site B:

```
ERROR: such policy already exists. anyway replace it: 192.168.32.0/24[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.32.1/32[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.32.0/24[0] 192.168
ERROR: such policy already exists. anyway replace it: 192.168.30.0/24[0] 192.168
[ToSiteA]: INFO: respond new phase 1 negotiation: 172.16.3.41[500]<=>172.16.0.40
INFO: begin Identity Protection mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
```

```
[ToSiteA]: INFO: ISAKMP-SA established 172.16.3.41[500]-172.16.0.40[500] spi:cc
[ToSiteA]: INFO: respond new phase 2 negotiation: 172.16.3.41[500]<=>172.16.0.4
[ToSiteA]: INFO: IPsec-SA established: ESP 172.16.0.40[0]->172.16.3.41[0] spi=2
[ToSiteA]: INFO: IPsec-SA established: ESP 172.16.3.41[500]->172.16.0.40[500] s
```

### Successful Aggressive Mode Tunnel

Log output from Site A:

```
[ToSiteB]: INFO: IPsec-SA request for 172.16.3.41 queued due to no phase1 found
[ToSiteB]: INFO: initiate new phase 1 negotiation: 172.16.0.40[500]<=>172.16.3.
INFO: begin Aggressive mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
NOTIFY: couldn't find the proper pskey, try to get one by the peer's address.
[ToSiteB]: INFO: ISAKMP-SA established 172.16.0.40[500]-172.16.3.41[500] spi:fc
[ToSiteB]: INFO: initiate new phase 2 negotiation: 172.16.0.40[500]<=>172.16.3.
[ToSiteB]: INFO: IPsec-SA established: ESP 172.16.3.41[0]->172.16.0.40[0] spi=1
[ToSiteB]: INFO: IPsec-SA established: ESP 172.16.0.40[500]->172.16.3.41[500] s
```

Log output from Site B:

```
[ToSiteA]: INFO: respond new phase 1 negotiation: 172.16.3.41[500]<=>172.16.0.4
INFO: begin Aggressive mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
NOTIFY: couldn't find the proper pskey, try to get one by the peer's address.
[ToSiteA]: INFO: ISAKMP-SA established 172.16.3.41[500]-172.16.0.40[500] spi:fc
[ToSiteA]: INFO: respond new phase 2 negotiation: 172.16.3.41[500]<=>172.16.0.4
[ToSiteA]: INFO: IPsec-SA established: ESP 172.16.0.40[0]->172.16.3.41[0] spi=1
[ToSiteA]: INFO: IPsec-SA established: ESP 172.16.3.41[500]->172.16.0.40[500] s
```

## Failed Connection Examples

These examples show failed connections for varying reasons. Particularly interesting parts of the log entries will be emphasized.

### Mismatched Phase 1 Encryption

Log output from Site A:

```
[ToSiteB]: INFO: IPsec-SA request for 172.16.3.41 queued due to no phase1 found
[ToSiteB]: INFO: initiate new phase 1 negotiation: 172.16.0.40[500]<=>172.16.3.
INFO: begin Identity Protection mode.
[ToSiteB]: ERROR: phase2 negotiation failed due to time up waiting for phase1.
INFO: delete phase 2 handler.
ERROR: phase1 negotiation failed due to time up. 96f516ded84edfca:0000000000000
```

Log output from Site B:

```
[ToSiteA]: INFO: respond new phase 1 negotiation: 172.16.3.41[500]<=>172.16.0.4
INFO: begin Identity Protection mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
ERROR: rejected enctype: DB(prop#1:trns#1):Peer(prop#1:trns#1) = 3DES-CBC:AES-C
ERROR: no suitable proposal found.
ERROR: failed to get valid proposal.
ERROR: failed to pre-process packet.
ERROR: phase1 negotiation failed.
```

In this case, the log entry tells you exactly what the problem was: This side was set for 3DES encryption, and the remote side is set for AES. Set both to matching values and then try again.

## Mismatched Phase 1 DH Group

In this instance, the log entries will be exactly as above, except that the emphasized line will instead be:

```
ERROR: rejected dh_group: DB(prop#1:trns#1):Peer(prop#1:trns#1) = 768-bit MODP
```

This error can be corrected by setting the DH group setting on both ends of the tunnel to a matching value.

## Mismatched Pre-shared Key

A mismatched pre-shared key can be a little tougher to diagnose. An error stating the fact that this value is mismatched is not printed in the log, instead you will see a message such as this:

```
[ToSiteB]: NOTIFY: the packet is retransmitted by 172.16.3.41[500] (1).
[ToSiteB]: ERROR: phase2 negotiation failed due to time up waiting for phase1.
```

If you notice an error similar to the above, check that the pre-shared keys match up on both ends.

## Mismatched Phase 2 Encryption

Log output from Site A:

```
[ToSiteB]: INFO: IPsec-SA request for 172.16.3.41 queued due to no phase1 found
[ToSiteB]: INFO: initiate new phase 1 negotiation: 172.16.0.40[500]<=>172.16.3.
INFO: begin Identity Protection mode.
INFO: received Vendor ID: DPD
INFO: received broken Microsoft ID: FRAGMENTATION
[ToSiteB]: INFO: ISAKMP-SA established 172.16.0.40[500]-172.16.3.41[500] spi:19
[ToSiteB]: INFO: initiate new phase 2 negotiation: 172.16.0.40[500]<=>172.16.3.
ERROR: fatal NO-PROPOSAL-CHOSEN notify message, phase1 should be deleted.
```

Log output from Site B:

```
[ToSiteA]: INFO: respond new phase 1 negotiation: 172.16.3.41[500]<=>172.16.0.4
INFO: begin Identity Protection mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
[ToSiteA]: INFO: ISAKMP-SA established 172.16.3.41[500]-172.16.0.40[500] spi:19
[ToSiteA]: INFO: respond new phase 2 negotiation: 172.16.3.41[500]<=>172.16.0.4
WARNING: trns_id mismatched: my:AES peer:3DES
ERROR: not matched
ERROR: no suitable policy found.
ERROR: failed to pre-process packet.
```

In these log entries, you can see that phase 1 completed successfully ("ISAKMP-SA established") but failed during phase 2. Furthermore, it states that it could not find a suitable proposal, and from the Site B logs we can see that this was due to the sites being set for different encryption types, AES on one side and 3DES on the other.

## Other Mismatched Phase 2 Information

Some other phase 2 errors such as mismatched PFS values or mismatched remote subnets will result in the same log output. In this case, there is little recourse but to check each option to ensure settings match up on both sides.

Log output from Site A:

```
[ToSiteB]: INFO: IPsec-SA request for 172.16.3.41 queued due to no phase1 found
[ToSiteB]: INFO: initiate new phase 1 negotiation: 172.16.0.40[500]<=>172.16.3.4
INFO: begin Identity Protection mode.
INFO: received Vendor ID: DPD
INFO: received broken Microsoft ID: FRAGMENTATION
[ToSiteB]: INFO: ISAKMP-SA established 172.16.0.40[500]-172.16.3.41[500] spi:2a
[ToSiteB]: INFO: initiate new phase 2 negotiation: 172.16.0.40[500]<=>172.16.3.4
[ToSiteB]: ERROR: 172.16.3.41 give up to get IPsec-SA due to time up to wait.
```

Log output from Site B:

```
[ToSiteA]: INFO: respond new phase 1 negotiation: 172.16.3.41[500]<=>172.16.0.40
INFO: begin Identity Protection mode.
INFO: received broken Microsoft ID: FRAGMENTATION
INFO: received Vendor ID: DPD
[ToSiteA]: INFO: ISAKMP-SA established 172.16.3.41[500]-172.16.0.40[500] spi:2a
[ToSiteA]: INFO: respond new phase 2 negotiation: 172.16.3.41[500]<=>172.16.0.40
ERROR: no policy found: 192.168.30.0/24[0] 192.168.32.0/24[0] proto=any dir=in
ERROR: failed to get proposal for responder.
ERROR: failed to pre-process packet.
```

The errors indicate that the proposals for phase 2 did not agree, and all values in the phase 2 section should be checked as well as the remote subnet definitions.

### Note

In some cases, if one side has PFS set to **`off`**, and the other side has a value set, the tunnel will still establish and work. The mismatch shown above may only be seen if the values mismatch, for example **`1`** vs. **`5`**.

## Other Common Errors

Some error messages may be encountered in the IPsec logs. Some are harmless, and others are indicative of potential problems. Usually the log messages are fairly straightforward in meaning, and indicate various problems establishing a tunnel with reasons why. There are some, however, that are a little more obscure.

```
Feb 20 10:33:41  racoon: [172.16.0.40] ERROR: failed to pre-process ph2 packet
Feb 20 10:33:41  racoon: ERROR: failed to get sainfo.
```

This is most commonly seen when the local and/or remote subnet definitions are incorrectly specified, especially if the subnet mask is set incorrectly on one side.

```
racoon: ERROR: none message must be encrypted.
```

Indicates that there may be a problem with traffic arriving from the opposite end of the tunnel. Try restarting the **racoon** service on the far side router by browsing to Status → Services and clicking **`Restart`** next to **racoon**.

```
racoon: ERROR: can't start the quick mode, there is no ISAKMP-SA.
```

May indicate a problem with sending local traffic to the remote tunnel, because an ISAKMP Security Association has not been found. It may be necessary to restart the **racoon** service on one or both sides to clear up this problem.

```
racoon: INFO: request for establishing IPsec-SA was queued due to no phase1 found
```

This is normal, and usually seen when a tunnel is first established. The system will first try to complete a phase 1 connection to the far side and then continue.

```
racoon: INFO: unsupported PF_KEY message REGISTER
```

This is harmless as well, and is typically found in the log shortly after the **racoon** daemon starts.

## Advanced debugging

When negotiation is failing, especially when connecting to third party IPsec devices where it isn't as easy to completely match the settings between the two sides, sometimes the only way to get adequate information to resolve the problem is to run **racoon** in debug mode. To do so, go to, System → Advanced on the Miscellaneous tab, and check Start racoon in debug mode. The debug information will be logged to the regular IPsec log.

If there is too much log information, you may need to run **racoon** in the foreground in debug mode from the shell. To do this, first log into your firewall using SSH and chose option **8** at the console menu for a command prompt. Run the following commands.

```
# killall racoon
```

Now wait about 5 seconds for the process to shut down, and launch it again using the following command:

```
# racoon -F -d -v -f /var/etc/racoon.conf
```

The first line stops the existing **racoon** process. The second starts **racoon** in the foreground (`-F`), with debugging (`-d`), increased verbosity (`-v`), using configuration file `/var/etc/racoon.conf` (`-f`). Running it in the foreground causes it to display its logs in your SSH session, so you can watch what is happening in real time. To quit out of **racoon**, press **Ctrl-C** and the service will be stopped. After finishing with debugging, you will need to start racoon normally. The easiest way to do this is to browse to Status → Services in the web interface and click next to **racoon**.

### Note

This method of debugging is disruptive to all IPsec on the system, when you kill off **racoon** you will drop all IPsec connections. Because of the volume of logs you will have to sort through with multiple IPsec connections enabled, while debugging a problem with one of them it's easier if you can disable the others while troubleshooting. Generally this method of debugging is only done when bringing up a new IPsec connection.

# Configuring Third Party IPsec Devices

You can connect any VPN device supporting standard IPsec with pfSense. It is being used in production in combination with numerous vendors' equipment, and should work fine with any IPsec capable devices in your network. 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. This section offers some general guidance on configuring IPsec VPNs with third party devices, as well as specific examples on configuring Cisco PIX firewalls and IOS routers.

## General guidance for third party IPsec devices

To configure an IPsec tunnel between pfSense and a device from another vendor, the primary concern is to ensure that your phase 1 and 2 parameters match on both sides. For the configuration options on pfSense, where it allows you to select multiple options you should usually only select one of those options and ensure the other side is set the same. The endpoints *should* 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 you believe are matching settings, and save and apply the changes on both sides.

Once you believe that the settings match on both ends of the tunnel, attempt to pass traffic over the VPN to trigger its initiation, then check your 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. You will find the pfSense side 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 your settings accordingly, as described in the section called “IPsec Troubleshooting”.

# Cisco PIX OS 6.x

The following configuration would be for a Cisco PIX running on 6.x as Site B from the example site-to-site configuration earlier in the chapter. See the section called “Site to site example configuration” for the Site A pfSense settings.

```
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 172.23.1.3 netmask 255.255.255.255 no-

!--- Phase 2
crypto ipsec transform-set 3dessha1 esp-3des esp-sha-hmac
access-list PFSVPN permit ip 10.0.10.0 255.255.255.0 192.168.1.0 255.255.255.0
crypto map dyn-map 10 ipsec-isakmp
crypto map dyn-map 10 match address PFSVPN
crypto map dyn-map 10 set peer 172.23.1.3
crypto map dyn-map 10 set transform-set 3dessha1
crypto map dyn-map 10 set security-association lifetime seconds 3600
crypto map dyn-map interface outside

!--- no-nat to ensure it routes via the tunnel
access-list nonat permit ip 10.0.10.0 255.255.255.0 192.168.1.0 255.255.255.0
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. See the section called “Site to site example configuration” for the corresponding Site A settings.

```
crypto isakmp enable outside

!--- Phase 1
crypto isakmp policy 10
  authentication pre-share
  encryption 3des
  hash sha
  group 2
  lifetime 86400

tunnel-group 172.23.1.3 type ipsec-l2l
```

```
tunnel-group 172.23.1.3 ipsec-attributes pre-shared-key aBc123%XyZ9$7qwErty99

!--- Phase 2
crypto ipsec transform-set 3dessha1 esp-3des esp-sha-hmac
access-list PFSVPN extended permit ip 10.0.10.0 255.255.255.0 192.168.1.0 255.25
crypto map outside_map 20 match address PFSVPN
crypto map outside_map 20 set peer 172.23.1.3
crypto map outside_map 20 set transform-set 3dessha1
crypto map outside_map interface outside

!--- no-nat to ensure it routes via the tunnel
access-list nonat extended permit ip 10.0.10.0 255.255.255.0 192.168.1.0 255.255
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. See section the section called “Site to site example configuration” for the Site A pfSense settings.

```
!--- Phase 1
crypto isakmp policy 10
  encr 3des
  authentication pre-share
  group 2
crypto isakmp key aBc123%XyZ9$7qwErty99 address 172.23.1.3 no-xauth

!--- Phase 2
access-list 100 permit ip 192.168.1.0 0.0.0.255 10.0.10.0 0.0.0.255
access-list 100 permit ip 10.0.10.0 0.0.0.255 192.168.1.0 0.0.0.255
crypto ipsec transform-set 3DES-SHA esp-3des esp-sha-hmac
crypto map PFSVPN 15 ipsec-isakmp
  set peer 172.23.1.3
  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 10.0.10.0 0.0.0.255 192.168.1.0 0.0.0.255
access-list 110 permit ip 10.0.10.0 0.0.0.255 any
route-map NONAT permit 10
  match ip address 110
```

# Chapter 18. OpenVPN

OpenVPN is an open source SSL VPN solution that can be used both for client remote access and site to site connectivity. OpenVPN supports clients on a wide range of operating systems including all the BSDs, Linux, Mac OS X, Solaris and Windows 2000 and newer. 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 primary location's 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.

There are several types of authentication methods that can be used with OpenVPN: shared key, X.509 (also known as SSL/TLS or PKI), user authentication via local, LDAP, and RADIUS, or a combination of X.509 and user authentication. For shared key, a single key is generated that will be used on both sides. SSL/TLS involves using a trusted set of certificates and keys. User authentication can be configured with or without SSL/TLS, but its use is preferable where possible due to the increased security is offers.

In this chapter, the settings for an OpenVPN instance are covered, as well as a run-through of the OpenVPN Remote Access Server wizard, client configurations, and examples of other connection scenarios.

Note that while OpenVPN is a SSL VPN, it is not a "clientless" SSL VPN in the sense that commercial firewall vendors commonly refer to it. You will need to install the OpenVPN client on all your 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, from the mailing list archives provides some excellent information: http://marc.info/?l=pfsense-support&m=121556491024595&w=2.

For general discussion of the various types of VPNs available in pfSense and their pros and cons, see Chapter 16, *Virtual Private Networks*.

# OpenVPN and Certificates

Using certificates is the preferred means of running remote access VPNs, because it allows you to revoke access to individual machines. With shared keys, you either have to create a unique server and port for each client, or distribute the same key 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 you otherwise wish to revoke the access of one person, you must re-issue the shared key to all clients. With a PKI deployment, if a client is compromised, or access needs to be revoked for any other reason, you can simply revoke that client's certificate. No other clients are affected.

In versions of pfSense prior to 2.0, the certificates had to be managed outside of the WebGUI. Starting with pfSense 2.0, the GUI now 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 router except for the default admin account. For further information on creating a certificate authority, certificates, and certificate revocation lists, see Chapter 8, *Certificate Management*.

# OpenVPN and IPv6

OpenVPN is the most complete VPN solution for IPv6 so far. You can connect an OpenVPN site-to-site tunnel to either an IPv4 address or an IPv6 address. You can pass IPv4 and IPv6 inside of an

OpenVPN tunnel. 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. As of this writing, the 2.3 client worked well, and some clients such as Viscosity were building their software on that version of the code.

# OpenVPN Configuration Options

This section describes all of the available options with OpenVPN and when you may want or need to use them. 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

This section describes each configuration option on the OpenVPN Server Edit screen.

### 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 it is compromised.

#### Peer to Peer (Shared Key)

A connection between local and remote networks that is secured by Shared Key. This choice is easier to setup, but is less secure. If a shared key is compromised, any router or client using that shared key will need to obtain a newly generated key.

#### Remote Access (SSL/TLS)

Historically the most common choice for OpenVPN, 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 your authentication is handled externally by LDAP or RADIUS.

#### Remote Access (SSL/TLS + User Auth)

This is the most secure choice that is 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

Select TCP or UDP here, 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. Unless there is a reason you must use TCP, such as the ability to bypass many firewalls by running an OpenVPN server on TCP port 443, you should use UDP. It is always preferable to use connectionless protocols when tunneling traffic. TCP is connection oriented, with guaranteed delivery. Any lost packets are retransmitted. This may sound like a good idea, but performance will degrade significantly on heavily loaded Internet connections, or those with consistent packet loss, because of the TCP retransmissions. You will frequently have TCP traffic within the tunnel. Where you have TCP wrapped around TCP, 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 lesser performance than if you were using UDP. You really do not want lost packets of encapsulated VPN traffic to be retransmitted. 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. In previous versions of pfSense, tun was assumed, and only a routed setup was possible. In pfSense 2.0 and above, there is a choice between the classic routed tun mode, and tap mode which is capable of either routing or bridging. The primary difference between the two is that tun works on OSI layer 3, while tap is capable of working at OSI layer 2. Not all clients support tap mode, using tun is recommended. 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 people.

## Interface

This lets you select which interface, VIP, or failover group that the OpenVPN server instance will listen upon for incoming connections, and also which interface the traffic from the server will exit. If you select a CARP type VIP, the OpenVPN instance will be stopped when the CARP VIP is in backup mode. This is done to prevent the backup unit from maintaining invalid routes or attempting to make outbound connections.

## Local port

The local port is the port number OpenVPN will use to listen. Your 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 your reference.

## Cryptographic Settings

This section controls how traffic to and from clients is encrypted and validated.

### Shared Key

When using a shared key instance, you can either check the Automatically generate a shared key box to make a new key, or uncheck that box to paste in a shared key from an existing OpenVPN tunnel. If you chose to generate 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 client and server. Check the box to Enable authentication of TLS packets if desired. If you do not have an existing TLS key, you may leave Automatically generate a shared TLS authentication key checked. If you have an existing key, uncheck that option and then paste it into the box that appears. If you chose to generate 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.

### Peer Certificate Authority

Here you must choose the certificate authority used to sign the server certificate for this OpenVPN server instance. If none appear in this list, you must 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 basically a list of certificates made from a given CA that should no longer be 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 this OpenVPN server instance. If none appear in this list, you must 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, if you are importing an existing OpenVPN configuration, you do not need to replicate the DH parameters from the previous server, a new set of DH parameters may be generated. The length of the desired DH parameters may be chosen from the drop-down box, either 1024, 2048, or 4096.

### Encryption algorithm

This is where you select 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. One common situation where you may want consider the cipher in depth this is when you are using a hardware crypto accelerator, such as `glxsb` built into ALIX hardware, or a `hifn` card. In these cases, you will see increased performance by using a hardware accelerated cipher. For ALIX or other hardware with `glxsb`, choose **`AES-CBC-128`**. For `hifn` hardware, chose any of the 3DES or AES options. See the section called “Hardware Crypto” for more information on using cryptographic accelerators.

## 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. You can choose any

subnet you like as long as it is not in use locally or at any remote site. You can fill in either an IPv4 Tunnel Network, an IPv6 tunnel network, both of them, or in the case of a tap bridge, neither.

For a site-to-site SSL/TLS server using IPv4, the IPv4 Tunnel Network size can alter how the server behaves. If you specify a *x.x.x.x/30* IPv4 Tunnel Network then it 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 you specify an IPv4 Tunnel Network larger than that, such as *x.x.x.x/24*, the server will accept multiple clients and can push settings, but does require **iroutes**. See the section called “Site to Site Example Configuration (SSL/TLS)” for more information on a site-to-multi-site example using a large tunnel network and **iroutes**.

## Redirect Gateway

Selecting this option will force all client-generated traffic to pass across the VPN tunnel, taking over as the client's default gateway.

## IPv4/IPv6 Local network

These fields specify what local networks should be reachable by VPN clients, if any. A route for these networks is pushed to clients connecting to this server. If you need to routes for more than one subnet of a particular family (IPv4 or IPv6), enter the subnets separated by a comma, e.g. *192.168.2.0/24, 192.168.56.0/24*. On versions older than pfSense 2.1, you can enter the first subnet here and see the section called “Custom configuration options” for information on adding the remaining subnets. 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 should always work for IPv6 provided a similar too-small mask isn't set.

## IPv4/IPv6 Remote Network

This option only appears if you are using a Peer to Peer type connection, and is not used for mobile clients. If a subnet is specified here, a route to this subnet via the other side of this OpenVPN connection will be added. If you need to add more than one Remote network subnet, enter the subnets separated by a comma, e.g. *192.168.2.0/24, 192.168.56.0/24*. On versions older than pfSense 2.1, enter the first here and see the section called “Custom configuration options” for information on adding the remaining subnets.

## Concurrent Connections

Here you may set how many clients may be simultaneously connected to this OpenVPN server instance at any given time.

## Compression

This check box enables LZO compression for your OpenVPN traffic. If this box is checked, the traffic crossing your 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 we suggest enabling this 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 your OpenVPN connection is already encrypted (such as SSH, SCP, HTTPS, amongst many other protocols), you should 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 your VPN traffic is almost entirely data that is already compressed.

## Type-of-Service

If this option is chosen, OpenVPN will set the 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, at the cost of some minor information disclosure.

### Inter-Client Communication

If clients need to communicate between each other, check this option. Without this option, they can only send traffic to the server (and any connected network for which they have a route).

### Duplicate Connections

By default, OpenVPN will associate an IP address from its tunnel network with a given certificate 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 mostly for security reasons, so the same certificate cannot be used by multiple people simultaneously. We recommend that a unique certificate be used for each connecting user. Otherwise, if a user's client is compromised, there is no way to revoke that one client alone, you would need to reissue certificates to all clients. If you must proceed with a set that uses the same certificate in multiple locations, check Duplicate Connections to allow the non-standard behavior of multiple clients using the same certificate.

## 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 your OpenVPN configuration. This allows connected clients to retain their connection if their IP changes. For clients on Internet connections where the IP changes frequently, or mobile users who commonly move between different Internet connections, you will need to check this option. Where the client IP is static or rarely changes, not using this option offers a minuscule security improvement.

### Address Pool

If you check this option, the server will assign a virtual adapter IP addresses to clients from the subnet specified by the Tunnel Network option. If you uncheck this option, IP addresses will not be assigned automatically, and clients will have to set their own static IP addresses manually in their client configuration files.

### Topology Subnet

By default, OpenVPN assumes a net30 topology for `tun` setups and a subnet topology for `tap` setups. With net30 on a `tun` setup, OpenVPN allocates a /30 CIDR network (four IPs, two usable) to each connecting client. In some cases, that is not desirable. The Topology Subnet option will make OpenVPN allocate only one IP per client, rather than an isolated subnet per client. This options is relevant only when supplying a virtual adapter IP address to clients when 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 this option, but there were significant fixes to it in OpenVPN 2.3 also, so using a current client is important. Some clients may break if this options is used, such as older versions of OpenVPN, Windows versions with older tun/tap drivers, or clients such as Yealink phones. Because it is not universally supported, this option is off by default and we recommend leaving this option unchecked unless you are certain all possible clients for this server will support it.

### DNS Default Domain

If you check this option, a field will appear where you may specify the DNS domain name to be assigned to clients. To ensure name resolution works properly for hosts on your local network where DNS name resolution is used, you should specify your internal DNS domain name here. For Microsoft Active Directory environments, this should usually be your Active Directory domain name.

### DNS servers

If you check this option, you may specify up to four DNS servers to be used by the client while connected to this server. For Microsoft Active Directory environments, this should specify your Active Directory DNS servers for proper name resolution and authentication when connected via OpenVPN.

### NTP servers

Checking this box will let you specify up to two NTP servers for syncing the time on clients. It can be an IP address or FQDN.

### NetBIOS Options

When the Enable NetBIOS over TCP/IP option is checked, several other NetBIOS and WINS related options will appear. If the box is unchecked, these settings will be disabled. The additional options are covered later.

#### 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** (broadcasts), **p-node** (point-to-point name queries to a WINS server), **m-node** (broadcast then query name server), and **h-node** (query name server, then broadcast).

#### 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 you to set up to two WINS servers to provide to clients for accessing and browsing NetBIOS resources by name across the VPN.

## Custom options

While the pfSense web interface supports all the most commonly used options, OpenVPN is very powerful and flexible and you may occasionally want or need to use options that are unavailable in the web interface. You can fill in these custom options here. These options are described further in the section called “Custom configuration options”.

# 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 allows you to configure all of the necessary prerequisites for an OpenVPN Remote Access Server: An authentication source, a Certificate Authority, a Server Certificate, and an OpenVPN server instance. By the end of the wizard, you should have a fully functioning sever, ready to accept connections from users. An example setup will be used to aide in explaining the options available in the wizard.

# Before Starting The Wizard

Before you start the wizard to configure your Remote Access Server, there are some details you need to plan beforehand.

## Determine an IP addressing scheme

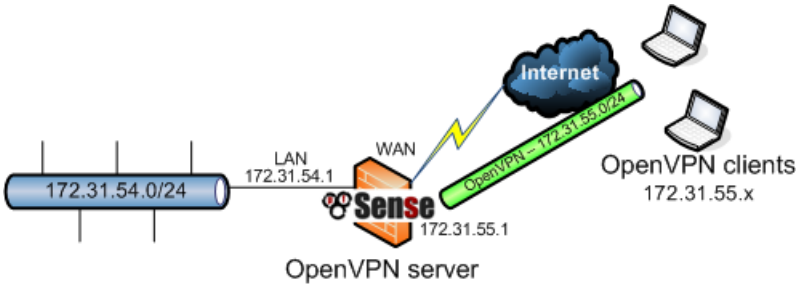
In addition to the internal subnets you will want clients to access, you need to choose an IP subnet to use for the OpenVPN connections. This is the subnet filled in under Tunnel Network in the server

configuration. Connected clients will receive an IP address within this subnet, and the server end of the connection also receives an IP on this subnet, where the client directs traffic for subnets routed through the OpenVPN connection. As always when choosing internal subnets for a single location, ideally this subnet should be CIDR summarizable with your internal subnets. The example network depicted here uses 172.31.54.0/24 for LAN, and 172.31.55.0/24 for OpenVPN. These two networks are summarized with 172.31.54.0/23, making routing easier to manage. CIDR summarization is discussed further in the section called “CIDR Summarization”.

## Example Network

Figure 18.1, “OpenVPN example remote access network” shows the network configured in this example.

**Figure 18.1. OpenVPN example remote access network**




# Choose Authentication Type

On the first screen of the OpenVPN Remote Access server wizard, you have to choose a method for user authentication. The choices available for Authentication Backend Type are Local User Access, LDAP, and RADIUS. If you have an existing authentication system in place, such as Active Directory, you may want to pick LDAP or RADIUS, depending on how that system is setup. You can choose Local User Access if you wish to manage the users, passwords, and certificates on the pfSense router. If LDAP or RADIUS are chosen, you cannot use per-user certificates without generating them externally. When using Local User Access, you can use per-user certificates 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.

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, press 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 our example, Local User Access will be chosen, but the other options are discussed for completeness as well.

# Choosing an LDAP Server

If there is an existing LDAP server defined on the pfSense system, you may choose it from the list. If you wish to use a different LDAP server, you may instead choose Add new LDAP server. If no LDAP servers are defined in pfSense, this step is skipped.

# Adding an LDAP Server

If no LDAP servers exist, or you chose to create a new LDAP server, a screen will be presented with the options needed to add a new server. Many of these options will depend on your specific LDAP directory configuration and structure. If you are unsure how to set a given value, consult your LDAP server administrator, software vendor, or documentation.

## Name

Descriptive name for this LDAP server, for your reference.

## Hostname or IP address

The hostname or IP address of the LDAP server. This server can be reachable on any interface, it does not have to be internal or directly connected.

## Port

The port on which the LDAP server may be contacted. The default port is **389** for standard TCP connections, and **636** for SSL, but depending on your LDAP server implementation and local configuration, this port may be different. Check with your LDAP administrator or software documentation to determine the proper port.

## Transport

The means by which an LDAP query will be made. This can be set to TCP - Standard for unencrypted connections, or SSL - Encrypted for secure connections. If your LDAP server is locally connected, you may prefer to use a standard connection. If the server is remote or crosses any untrusted network links, you may prefer to use SSL. You must also ensure that your LDAP server is configured to listen for the type of connection chosen.

## Search Scope Level

Selects how deep you want to search in the LDAP directory. You may choose One Level or Entire Subtree. Depending on the other parameters chosen, and the structure of your LDAP directory, you may want to restrict searches to a specific level.

## 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 the 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 your LDAP directory software and structure. Typically *cn* for OpenLDAP and Novell eDirectory, and *samAccountName* for Microsoft Active Directory.

## Group Naming Attribute

Varies depending on your LDAP directory software and structure, but is most typically *cn*.

### Member Naming Attribute

Varies depending on your 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 system, you may choose it from the list. If you wish to use a different RADIUS server, you may instead choose Add new RADIUS server. If no RADIUS servers are defined in pfSense, this step is skipped.

## Adding a RADIUS Server

If no RADIUS servers exist, or you chose to create a new RADIUS server, a screen will be presented with the options needed to add a new server. If you are unsure how to set a given value, consult your RADIUS server administrator, software vendor, or documentation.

### Name

Descriptive name for this RADIUS server, for your reference.

### Hostname or IP address

The hostname or IP address of the RADIUS server. This server can be reachable on any interface, it does not have to be internal or directly connected.

### Authentication Port

Port used by your RADIUS server for accepting Authentication requests, typically *1812*, but in some older RADIUS implementations may be *1645*.

### Shared Secret

The Shared Secret is the password configured on the RADIUS server for accepting authentication requests from the IP address of your pfSense router.

## Choosing a Certificate Authority

If there is an existing Certificate Authority defined on the pfSense system, you may choose it from the list. If you wish to use create a new Certificate Authority, you may instead choose Add new CA. If no Certificate Authorities are defined in pfSense, this step is skipped.

## Creating a Certificate Authority

This step allows you to create a new certificate authority (CA), and presents all of the options you need. Every option on this page is required, and all fields will need to be filled out correctly to proceed. The CA is used to establish a trust base from which your 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. For more information on creating and managing CAs, see the section called “Certificate Authority Management”.

### Descriptive Name

A name for your reference, to identify this certificate. This is the same as common-name field for other Certificates. For this CA, we'll call is *ExampleCoCA*. Although using spaces in this field is

acceptable, we strongly discourage using spaces in a Common Name field. The primary reason is that clients tend to have issues with such common names.

### Key Length

Size of the key which will be generated. The larger the key, the more security is 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 you do not know your two-letter ISO country code, you may find it here http://www.iso.org/iso/english_country_names_and_code_elements. Since our ExampleCo company is set in the United States, we'll enter *US*.

### State or Province

Full State of Province name, not abbreviated (e.g. Indiana, California). ExampleCo is located in *Kentucky*, so that is what will be entered.

### City

City or other Locality name (e.g. Indianapolis, Toronto). ExampleCo's headquarters is in *Louisville*.

### Organization

Organization name, often the Company or Group name. *ExampleCo* would go here. Be sure not to use special characters here, even punctuation.

### E-Mail

E-mail address for the Certificate contact. Often the e-mail of the person generating the certificate, such as *vpnadmin@example.com*.

## Choosing a Server Certificate

If there is an existing Certificate defined on the pfSense system, you may choose it from the list. If you wish to use create a new Certificate, you may instead choose Add new Certificate. If no Certificates are defined in pfSense, this step is skipped.

## Adding a Server Certificate

This screen allows you to create a new server certificate, 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 our example, the same information from the previous step is used. For more information on creating and managing certificates, see the section called “Certificate Management”.

### Descriptive Name

This is the Common Name (CN) field for the server certificate, and also used to reference the certificate in pfSense. Although using spaces in this field is acceptable, we strongly discourage using spaces in a Common Name field. The primary reason is that clients tend to have issues with such common names.

## 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.

## 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. Kentucky, Indiana, Ontario).

## City

City or other Locality name (e.g. Louisville, Indianapolis, Toronto).

## Organization

Organization name, often the Company or Group name. Be sure not to use special characters here, even punctuation.

## E-Mail

E-mail address for the Certificate contact. Often the e-mail of the person generating the certificate.

# Configuring OpenVPN Server Settings

This screen will 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 above in the section called “OpenVPN Configuration Options”, you may refer to that section for details about each field and how it should be set. Because the options have been explained in detail previously, only the settings for our example will be mentioned.

## General OpenVPN Server Information

These options control options specific to how the OpenVPN instance is run on this router.

### 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 you have an existing VPN, 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

Now we will set the encryption settings for the tunnel.

## TLS Authentication

We want to use TLS, so **check** Enable authentication of TLS packets.

## Generate TLS Key

We do not have an existing TLS key, so **check** Automatically generate a shared TLS authentication key.

## TLS Shared Key

Since we do not have an existing TLS key, leave this blank.

## DH Parameters Length

Select **2048**, as it should be a 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.

## Hardware Crypto

If your firewall device has a hardware cryptographic accelerator, such as a `hifn` card or the onboard `glxsb` on the ALIX platform, you may choose it here. Most accelerator boards hook in using the BSD cryptodev engine, so when in doubt, select that. This setting will allow OpenVPN to take advantage of the hardware acceleration. You must also be using a cryptography algorithm supported by your accelerator. For `glxsb`, that is only AES-128-CBC. Modern Hifn cards such as the Soekris vpn1411 support 3DES and 128, 192 and 256 bit AES.

# Tunnel Settings

This part covers how traffic coming from the remote clients will be routed.

## Tunnel Network

As in the diagram at the start of this example, the subnet **172.31.55.0/24** has been chosen for the VPN clients.

## Redirect Gateway

For ExampleCo's purposes, we only want traffic on the VPN which is destined for our subnets at the main office, so this box will be left **unchecked**.

## Local Network

This would be the main office subnet, which in our example is **172.31.54.0/24**.

## 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 box will be **checked**.

### Type-of-Service

This box will be **unchecked**, as there should be no traffic on this VPN which would need prioritization/QoS.

### Inter-Client Communication

Because the clients should have no need to connect to other client machines, this box will be **unchecked**.

### Duplicate Connections

Because we will have unique certificates for every client, this will remain **unchecked**.

## Client Settings

This part controls specific settings given to the connecting clients when a connection is established.

### Dynamic IP

Our clients will connect from all over the country and unknown networks, so their IP addresses are likely to change without notice, so this option should be **checked**.

### Address Pool

We want the clients to be assigned addresses from the tunnel network above, so this should be **checked**.

### DNS Default Domain

We will put the domain for ExampleCo here, **example.com**.

### DNS Servers

For a DNS server to provide the client, any internal DNS server could be used. ExampleCo has a Windows Active Directory Domain Controller, so it will be used here, **172.31.54.5**.

### NTP Servers

The server above will also be used to synchronize client PC clocks, so enter it here too: **172.31.54.5**.

### NetBIOS Options

Clients will need access to Windows shares behind the VPN, so we want to **check** Enable NetBIOS over TCP/IP.

### NetBIOS Node Type

Because we will also be using a WINS server and not broadcasts, select **p-node**.

### NetBIOS Scope ID

This will be left blank, since we do not wish to limit the NetBIOS scope.

### WINS Servers

The above Windows server is also a WINS server, so use it here too: **172.31.54.5**.

### Advanced

At this time no additional tweaks are needed, so this may be left blank.

# Firewall Rule Configuration

At with other parts of the firewall, by default all traffic is blocked from connecting to VPNs or passing over VPN tunnels. This screen lets you add 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 which lets clients connect. It allows all clients to connect by default, so if you intend on only allowing connections from a limited set of IPs or subnets, you can either make your own rule or check this box and change the rule it creates. Since in our example we have clients connecting from all over the country, the rule created by this checkbox is ideal, so we will **check** this box.

## Traffic from clients through VPN tunnel

This setting will allow all traffic to cross the OpenVPN tunnel, which is what we want to do for this example, so **check** this box.

# Finishing the Wizard

The wizard is now complete, and the tunnel should be fully configured and ready for client connections. From here, the next steps will be to add users, and configure client PCs. If you need to make any adjustments to the automatically generated firewall rules, now would be the time to do so.

# Configuring Users

At this point you should have a VPN tunnel configured but there are may not yet any clients which can connect. The method for adding users to your VPN will depend upon the authentication method chosen when creating the OpenVPN server instance. More details on adding users can be found in Chapter 7, *User Management and Authentication*. More information on managing user certificates can be found in the section called “User Certificates”.

# Local Users

To add a user that can connect to OpenVPN, you must add them under System → User Manager. Navigate to that page, then click to add a new user. Enter a Username, a Password and confirmation, fill in Full Name if you want. Check Click to create a user certificate, which will make some more options appear. Enter the VPN connection name or some other pertinent information into the Description field, choose the same Certificate Authority as used to create the OpenVPN server instance, then choose a Key Length, and enter a Lifetime (these two may be left at their defaults). To finish, click the Save button.

To view or change the user, click next to the row containing the user you want to see/edit. Near the bottom you can see there are two icons, which offer a description when the mouse is hovered above them. The first icon will download the private key for this certificate, and the second icon will download the actual certificate. Both are needed for the client software if you are creating a client configuration manually. You can skip this part if you will be using the OpenVPN Client Export Package, described in the section called “OpenVPN Client Export Package”. The client export package is a much easier way to download client configurations and installation files.

# LDAP or RADIUS Users

Adding LDAP and RADIUS users will fully depend on your server implementation and management tools, which are beyond the scope of this book. Contact your server administrator or software vendor for assistance. You cannot create certificates for LDAP or RADIUS users from within the firewall's

web interface in a way that reflects a user-certificate relationship. However, you can create the certificates on their own using the certificate manager.

# OpenVPN Client Installation

With the server configuration complete, OpenVPN now needs to be installed on the client system. The same OpenVPN installation can function as either a client or server, so there is only one installation routine. It functions as instructed in the configuration provided, which will be covered in the next section. This section provides an overview of installation on several common operating systems.

# OpenVPN Client Export Package

By far the easiest way to configure an OpenVPN client on Windows, Mac, or Android 4.x is to use the OpenVPN Client Export Package from your firewall. To install the client, browse to System → Packages, locate OpenVPN Client Export in the list, and click the button to install. Once installed, you will find it under VPN → OpenVPN, on Client Export tab.

To use the package, First pick your OpenVPN server instance from the Remote Access Server drop-down list. Next, for the Host Name Resolution option, pick how you want the "remote" entry formatted for the client. Typically, Interface IP Address is best for installations with a static IP on WAN, Installation Hostname may be used if desired but is especially useful if the server has a dynamic IP and uses dynamic DNS, or you may choose Other and manually enter the hostname or IP address for the client to use when connecting.

If your server certificate common name contains a space, and you have a client that requires that such a CN be quoted in the client configuration file, check Quote Server CN.

Under Certificate Export Options, you can use the checkbox for Use Microsoft Certificate Storage instead of local files if desired, and if you check Use a password to protect the pkcs12 file contents and enter a Password, the certificates and keys supplied to the client will be protected with a password. If your OpenVPN server is setup for user authentication, this will give your users two different password prompts when loading the client. One to decrypt the keys and certificates, and another upon connecting.

If you know the client will be located behind an HTTP proxy, you can check Use HTTP proxy to communicate with the server, and then supply an IP Address, Port, and HTTP Proxy Authentication type if needed. These settings will be preconfigured on the client.

If you need to pass any additional configuration options to the client, you may do so in the Additional configuration options box. This is roughly equivalent to the Advanced options box on the OpenVPN instance configuration screens, but from the perspective of the client.

Under Client Install Packages you will see a list of users on the system which have associated certificates created. If no users are listed, you must create them on the system as described in the section called “Local Users”. Next to each user, there are several choices for exporting client settings. If you expect to see a client here but do not, it may be that the client certificate was not generated against the same CA as the OpenVPN server.

## Standard Configurations

Under this heading, the Archive option 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 good for use with Linux clients or Tunnelblick.

The File Only choice will download only the 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 will download a single configuration file with the certificates and keys inline. This format is ideal for use on Android and iOS clients, or for manually copying a configuration to a system that

already has a client installed. This option should work for any client type so long as it's based on OpenVPN version 2.1 or newer.

The Android choice is meant to be used with the Android OpenVPN client mentioned in the section called “Android 4.x”. The OpenVPN Connect (iOS/Android) option downloads a configuration file meant for the OpenVPN Connect client, described in the section called “iOS”. Lastly, the Others link downloads an inline configuration file in a format that should be usable by any standard OpenVPN client on platforms such as Windows or BSD/Linux. It also works well with Tunnelblick on OSX, simply download the inline config and drag it into Tunnelblick's configurations folder.

## SIP Phone archives

If the OpenVPN server is configured as SSL/TLS only — no authentication — more 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.

## Windows Installers

This simple option will download an EXE file which contains the OpenVPN installer with embedded configuration files. The installer runs just like the normal Windows OpenVPN client installer, but will also install all of the settings needed. Please see the section called “Windows Installation” below for some notes on how to install and run the Windows client. Currently, there are two options available, 2.2 and 2.3. The 2.3 version works better on Windows Vista/7, and in some cases can eliminate the need to run the client as Administrator.

### 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 you may be left with the client installed but no imported configuration.

On Windows Vista and Windows 7 with UAC (User Account Control) enabled, you must 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. You may also adjust the properties of the shortcut to always launch the program as Administrator. This option is found on the Compatibility tab of the shortcut properties. The 2.3 client improves on this behavior, but in some cases may still require Administrator access. One way around that requirement is to check OpenVPNManage before exporting to use an alternate OpenVPN management GUI on Windows.

## Viscosity Bundle

This works like the configuration archive above, but is for the Viscosity OpenVPN client used in OSX. If you have the Viscosity client already installed, you may download this bundle, and click it to import it into the client.

# Client Installation

If you choose to perform a manual client installation, instead of using the OpenVPN Client Export Package mentioned above, there are more steps involved in getting the software and settings onto the client PCs.

## Windows Installation

The OpenVPN project provides an installer for Windows 2000 through Windows 7, downloadable from http://openvpn.net/index.php/open-source/downloads.html. At the time of this writing, the best version for most Windows users is 2.3. The 2.3 series is still in Beta, but several other clients like Viscosity are shipping based on 2.3 due to its increased support for IPv6. In our testing it has performed well. The current stable 2.2 version is a good alternative for those who do not need the extra features or do not wish to run a Beta client. The installation is straight forward, just accept all the defaults. The installation will create a new Local Area Connection on your system for the `tun` interface. This interface will be connected when the VPN is connected, and otherwise show as disconnected. No configuration of this interface is necessary, as its configuration will be pulled from the OpenVPN server.

### Note

On Windows Vista and Windows 7 with UAC (User Account Control) enabled, you must 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. You may also adjust the properties of the shortcut to always launch the program as Administrator. This option is found on the Compatibility tab of the shortcut properties. The 2.3 client improves on this behavior, but in some cases may still require Administrator access. One way around that requirement is to check OpenVPNManage before exporting to use an alternate OpenVPN management GUI on Windows.

## Mac OS X Clients and Installation

There are three client options for Mac OS X. One is the simple OpenVPN command line client. Most users prefer a graphical client, and there are two options available for OS X. Tunnelblick is a free option available for download at http://www.tunnelblick.net. I have used it in the past with success. Another GUI option is the commercial Viscosity client available at http://www.viscosityvpn.com. At the time of this writing, it costs $9 USD for a single seat. If you use OpenVPN 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.

## FreeBSD Installation

If you have a stock FreeBSD installation, you can find OpenVPN in ports. To install, just run:

```
# cd /usr/ports/security/openvpn && make install clean
```

## Linux Installation

Linux installation will vary depending on your 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 your distribution of choice and "**`installing OpenVPN`**" to find information.

## Android 4.x

As of Android 4.0 (Ice Cream Sandwich, ICS), there is a VPN API that allows Android to run an OpenVPN client without root privileges. For devices running ICS, Android 4.1 (Jelly Bean), or newer, there is a free OpenVPN app in the Google Play store that works excellently in all of our tests. It is called OpenVPN for Android [https://play.google.com/store/apps/details?id=de.blinkt.openvpn] by Arne Schwabe. There are other OpenVPN clients out there, but most require rooting your Android device, which will let OpenVPN work on older versions of Android, but is far outside the scope of this book.

You can use the OpenVPN Client Export package on pfSense to export an Android type Inline Configuration, and then transfer the resulting `.ovpn` file to the target device. You can copy it directly, e-mail it to yourself, etc.

Open the OpenVPN app, and click All your precious VPNs. Then click Import (File folder icon in top right), find the `.ovpn` file you saved above and click it. Click Select, then click the Save icon.

Now that it's saved, you need to tell it your username if you're using a User Auth type. In the list of VPNs, click the icon to edit the VPN (looks like three sliders). Click Edit in the top bar (Pencil icon). Click Basic, and Fill in the Username. Click back repeatedly until you get back to the VPN list.

You should now be able to connect to the VPN.

### Note

The OpenVPN Connect [https://play.google.com/store/apps/details?id=net.openvpn.openvpn] 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.

## Android 2.1 through 3.2

For users of Android 2.1 through 3.2, there is a non-root OpenVPN client called FEAT VPN [https://play.google.com/store/apps/details?id=com.featvpn.app.lite]. Some users have reported success with using it in places where it is not possible to run Android 4.x or root the device.

## iOS

iOS is also capable of running OpenVPN natively, now without needing to jailbreak your iOS device. The iOS client is called OpenVPN Connect [https://itunes.apple.com/us/app/openvpn-connect/id590379981] and is available in the App Store. The app must have the config file and certificates configured outside of the iOS device and then imported to it. You can use the OpenVPN Client Export package on pfSense to export an OpenVPN Connect type Inline Configuration, and then transfer the resulting `.ovpn` file to the target device. You can then use iTunes to transfer the files into the app or e-mail it to yourself.

If you e-mail the configuration file to yourself, open the Mail app on your device and then open the e-mail message containing the attachment. If you have the OpenVPN Connect app installed already, when you tap on the attachment, one of the choices will be to open it with the app. Tap that, and it should offer to import the configuration. Click the + button and the profile should import cleanly.

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.

Using iTunes to transfer the configuration to the iOS device is rather simple. First, export the OpenVPN Connect type Inline Configuration file for the VPN. Next, connect your iOS device to your computer and open iTunes, find and install the OpenVPN Connect app. Inside of iTunes, click the device icon in the toolbar, then click on the Apps tab to open the list of apps on the device. At the bottom of this screen is a section titled File Sharing and under that heading, is an icon for OpenVPN, click that icon and a list of files will show on the right under the heading OpenVPN Documents. Use your operating system's file manager to drag and drop the `.ovpn` file into this area, or click Add and locate the file that way. The file should be immediately available on the iOS device. Open the OpenVPN Connect app and it should offer to import the profile. Click the + button, and the profile should import without issue.

If you configured the profile for user authentication, it will prompt you for the credentials and you may optionally save them. Underneath the credentials is a connection status, which will change between

Disconnected and Connected and also indicate when a connection is being attempted. Clicking this will take you to the OpenVPN client log which is very useful if you encounter connection problems.

To connect the VPN, move the slider at the bottom of the profile from Off to On, and it will attempt to connect. To manually disconnect, move the slider back to Off.

This OpenVPN client does not fully support IPv6 in our current configuration, but may in the future. It has some other limitations, such as not supporting `tap` mode or the tls-remote option, but ultimately it works well for what most people need. If you need to use IPv6 with this client, be sure to check Topology Subnet on the OpenVPN server instance settings.

If you are manually building a configuration file for this client, it requires either an inline configuration style or separate CA, client certificate, client certificate key, and if you used it, TLS key files. 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 config they extracted from the Viscosity bundle.

If you attempt to import a previously exported inline style config and encounter errors, first you must edit the configuration file and remove any lines containing **`[inline]`** and also **`tls-remote`**. It should then be possible to import the configuration file.

# Client Configuration

After installing OpenVPN, you need to copy the certificates to the client and create the client configuration file.

## Copy certificates

Three files from your firewall are needed for each client: the CA certificate, the client certificate, and the client key. The CA certificate can be exported and saved from System → Cert Manager on the CAs tab, save this as `ca.crt`. The client's certificate and key can be downloaded as described in the section called “Local Users”, save these as *`username`*`.crt` and *`username`*`.key`. Now copy these files to the OpenVPN `config` directory on the client. If you are using TLS authentication on this OpenVPN server, copy and paste the TLS key from the server configuration screen 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 openvpn.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
```

The **remote** line specifies the host and port of the remote OpenVPN server. An IP address or FQDN can be specified here. The **proto** line specifies the protocol used by the OpenVPN connection. Change this line to **proto tcp** if you chose TCP rather than UDP for your OpenVPN server. The **ca**, **cert**, and **key** lines need to be modified accordingly for each client. If you are not using TLS authentication, you may omit the tls-auth line. If you are not using a remote access setup that includes passwords, you may omit the auth-user-pass line also.

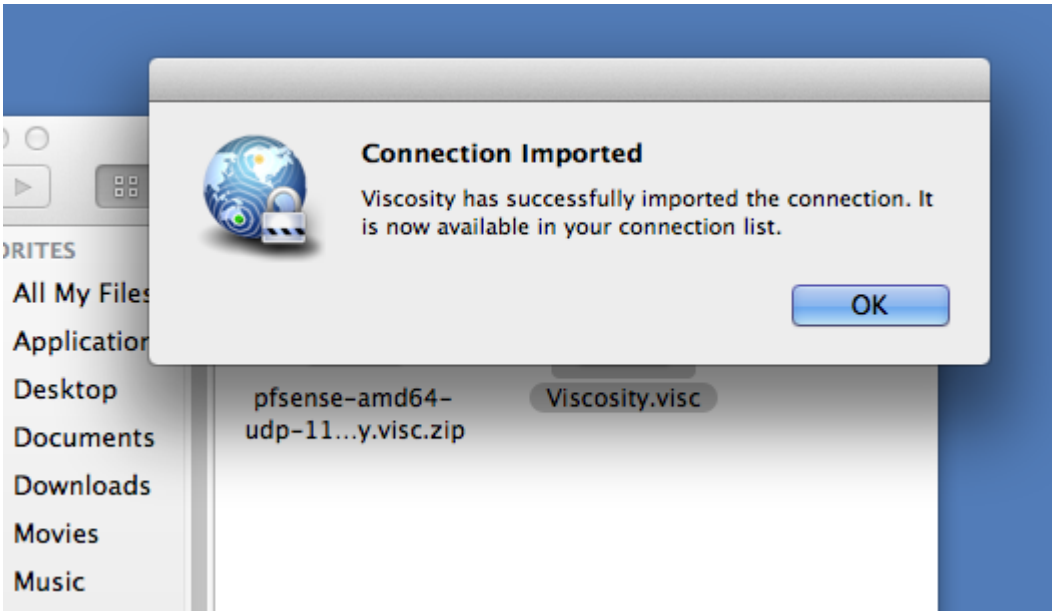
### Distributing configuration and keys to clients

The easiest way to distribute the keys and OpenVPN configuration to clients is via the OpenVPN Client Export package. If you are unable to use that package, you can place the needed files in a ZIP archive, or self-extracting ZIP automatically extracting to `C:\Program Files\OpenVPN\config`. This must be transmitted securely to the end user, and should never be passed over untrusted networks unencrypted.

## Configuring Viscosity

When using the Viscosity client, you can configure it manually or use the OpenVPN Client Export package to import the configuration. Viscosity provides a GUI configuration tool that can be used to generate the underlying OpenVPN client configuration if you like. You can import the CA and certificates manually, and set all of the parameters by hand. However, we will cover importing a Viscosity bundle from the export package since it is much simpler. First, download a copy of the Viscosity bundle for your client from the OpenVPN Client Export package to a folder of your choosing on the Mac. The file downloaded will end in `.visc.zip` indicating that it is a compressed archive. Double click this file, and it will expand to `Viscosity.visc`. Then double click `Viscosity.visc` and Viscosity will open and import the connection, as shown in Figure 18.2, “Viscosity Import”.

**Figure 18.2. Viscosity Import**




Afterwards, you can delete the `Viscosity.visc` directory and the `.zip` archive. Ensure this folder is kept secure, or has the files deleted once finished configuring Viscosity. After importing your configuration, Viscosity will be running, and may be found in the menu bar.

Click the lock icon added to the menu bar at the top of the screen, and click Preferences to check that the configuration was imported as shown in Figure 18.3, “Viscosity Preferences”.

**Figure 18.3. Viscosity Preferences**

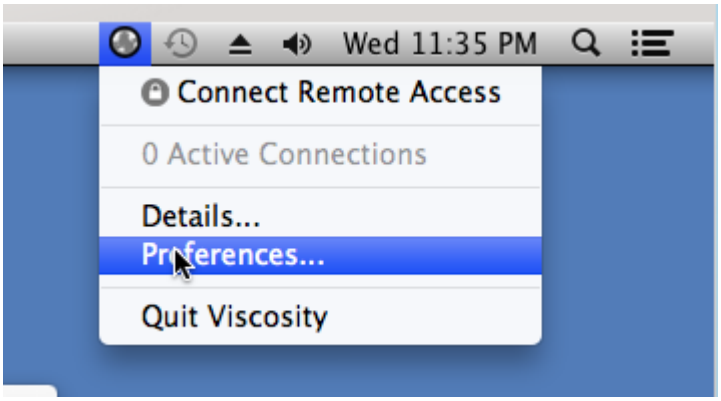
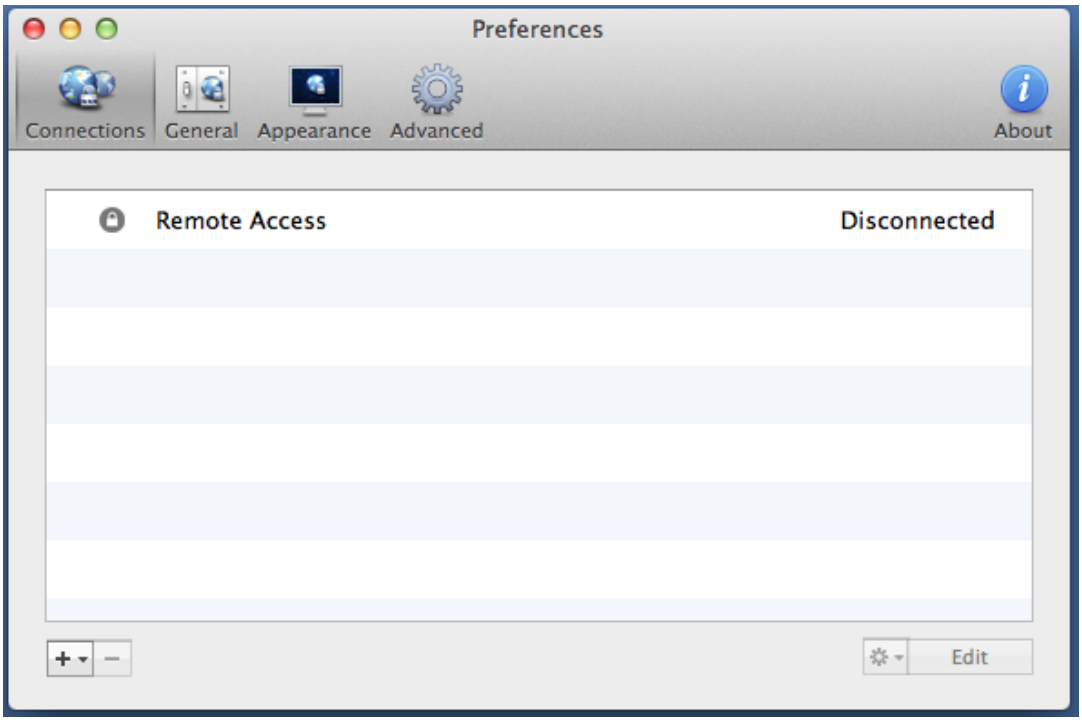
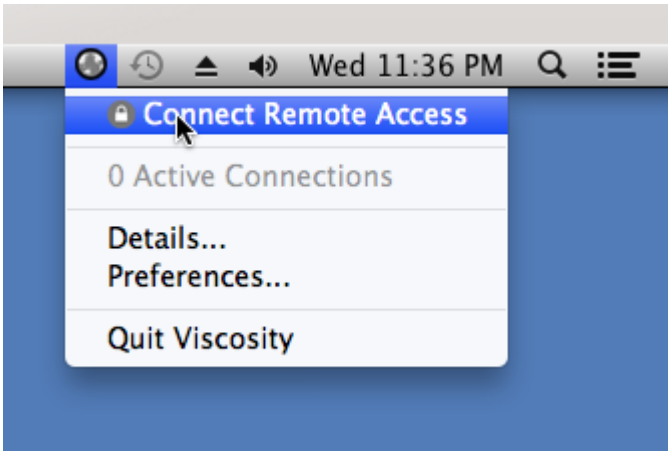



**Figure 18.4. Viscosity View Connections**



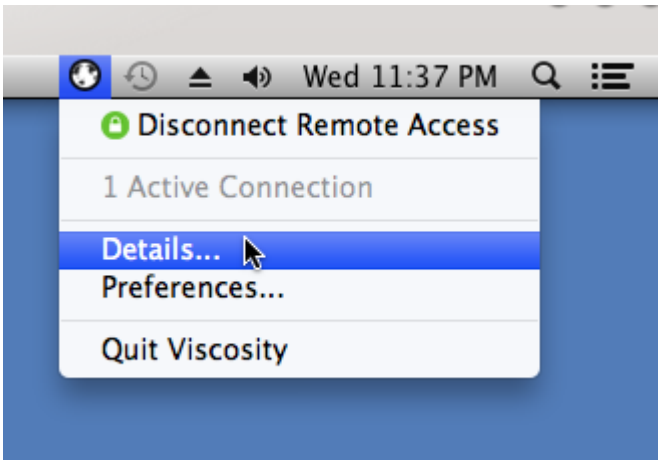

If the connection imported successfully, you will see it listed under Connections as shown in Figure 18.4, “Viscosity View Connections”. Close the Preferences screen, then click the lock in the menu bar, and the name of your VPN connection to connect, as shown in Figure 18.5, “Viscosity connect”.

**Figure 18.5. Viscosity connect**



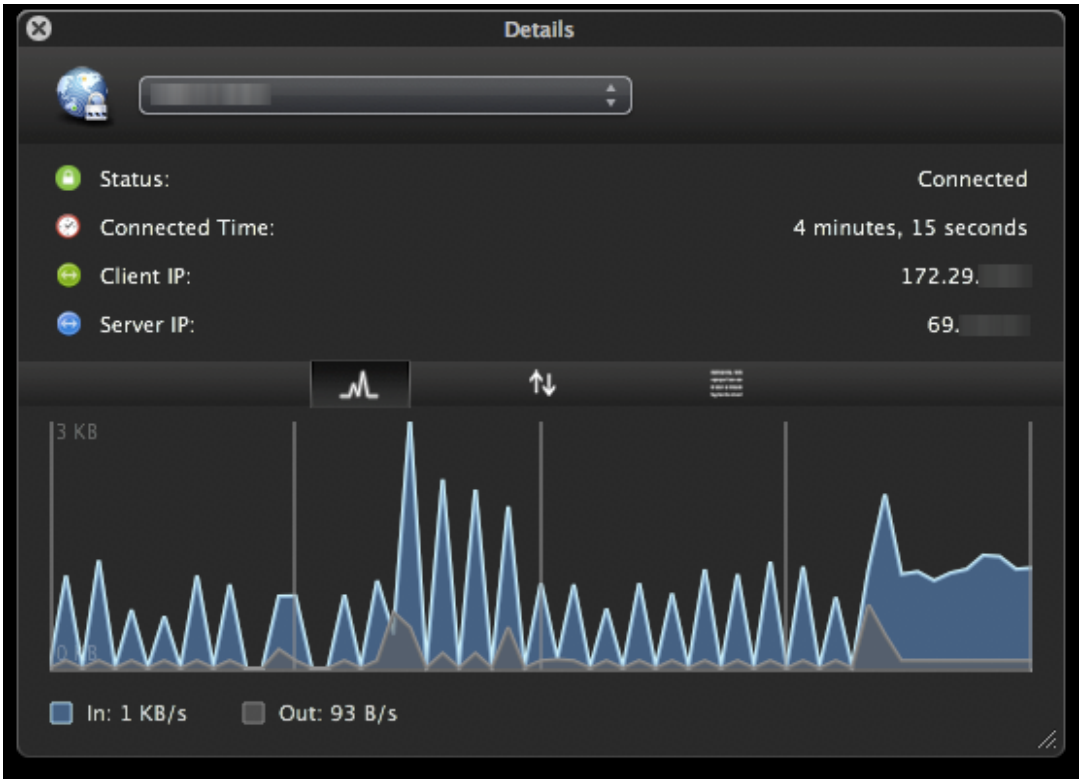

After a few seconds, the lock in the menu bar should turn green to show it connected successfully. By clicking on it, and clicking Details as shown in Figure 18.6, “Viscosity menu”, you can see information on the connection.

**Figure 18.6. Viscosity menu**



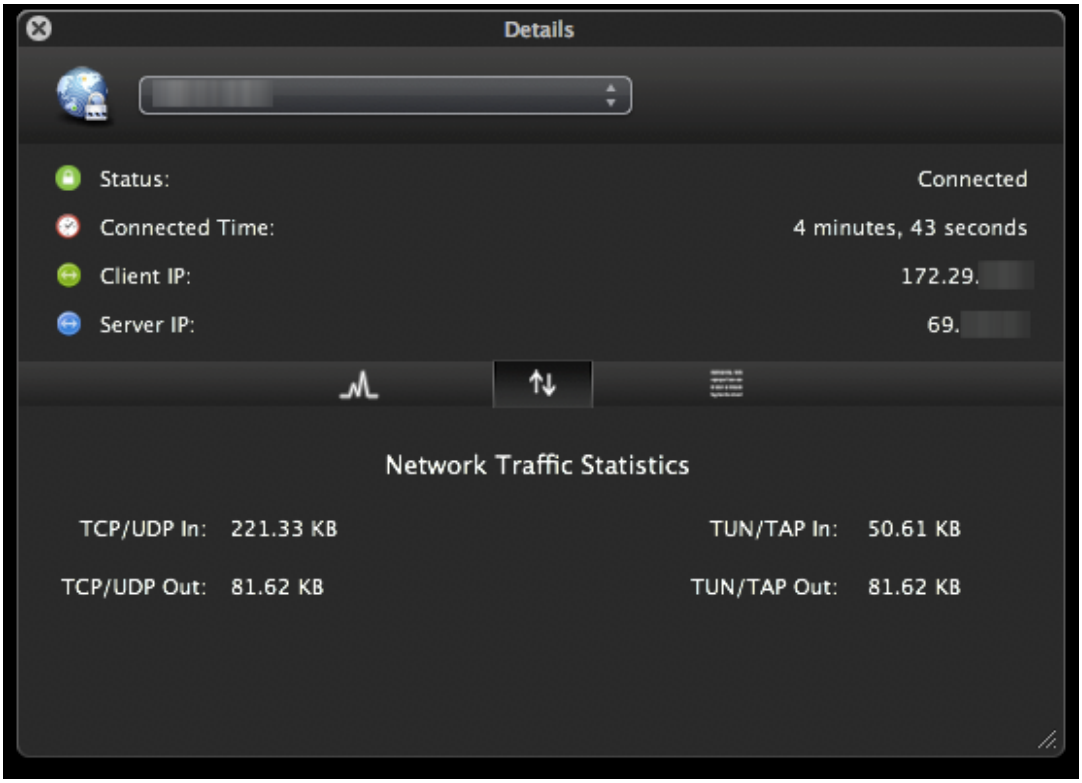

On the first screen (Figure 18.7, “Viscosity details”), you see the connection status, connected time, the IP assigned to the client, and the IP of the server. A bandwidth graph is displayed at the bottom of the screen, showing the throughput in and out of the OpenVPN interface.

**Figure 18.7. Viscosity details**




By clicking on the up/down arrows button in the middle of the details screen, you can see further 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 18.8, “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.

**Figure 18.8. Viscosity details: Traffic Statistics**

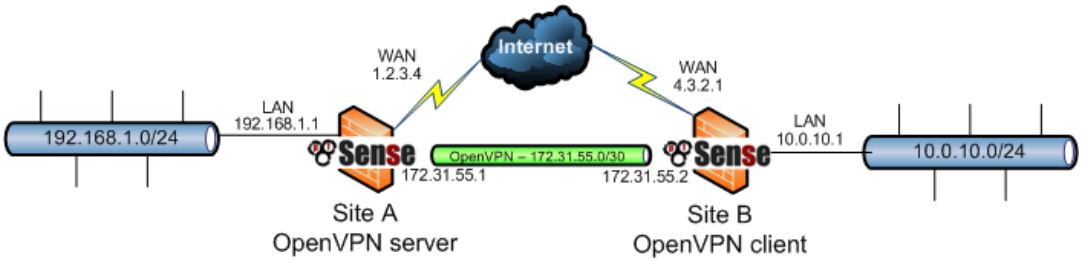



Clicking on the third icon in the middle of the Details screen shows the OpenVPN log file (Figure 18.9, “Viscosity details: Logs”). If you have any trouble connecting, review the logs here to help determine the problem. See also the section called “Troubleshooting OpenVPN”.

**Figure 18.9. Viscosity details: Logs**

# Site to Site Example Configuration (Shared Key)

**Figure 18.10. OpenVPN example site to site network**




This section describes the process of configuring a site to site connection using shared keys. When configuring a site to site OpenVPN connection, one firewall will be the server and the other will be the client. Usually your main location will be the server side and the remote offices will act as clients, though the opposite is functionally equivalent. In addition to the subnets on both ends, as with the remote access OpenVPN configuration, there will be a dedicated subnet in use for the OpenVPN interconnection between networks. The example configuration described here is depicted in Figure 18.10, “OpenVPN example site to site network”.

**`172.31.55.0/30`** is used as the Tunnel Network. The OpenVPN tunnel between the two firewalls gets an IP 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

Browse to VPN → OpenVPN and click on the Server tab. The following fields are configured, with everything else left at defaults.

| | |
|---|---|
| Server Mode | Select **`Peer to Peer (Shared Key)`**. |
| Description | Enter something here to describe the connection. |
| Shared key | Check Automatically generate a shared key, or you may paste in a pre-existing shared key for this connection here. |
| Tunnel Network | Enter **`172.31.55.0/30`** here. |
| Remote network | Enter **`10.0.10.0/24`** here. |

That is everything that must be configured for the OpenVPN server to function in this scenario. Click Save.

You will need to copy the shared key which was just generated, for use on the client system. From the list of OpenVPN server instances, click next to the one that was just created. Find the Shared Key box and select all of the text inside, then copy the text to the clipboard. You can either save this to a file, or paste it into a text editor such as Notepad temporarily.

Next you will need to add a firewall rule on WAN allowing access to the OpenVPN server. Specify the protocol **`UDP`**, source IP as the client's IP address if it has a static IP, or **`any`** if its IP is dynamic. Destination is the **`WAN Address`**, and destination port is **`1194`** in this instance. Figure 18.11, “OpenVPN example site to site WAN firewall rule” shows the firewall rule used for this example.

**Figure 18.11. OpenVPN example site to site WAN firewall rule**

Apply changes after the firewall rule is added, and the server configuration is finished.

## Configuring Client Side

On the client end, browse to VPN → OpenVPN and click on the Client tab. The following fields are configured, 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. |
| Description | Enter something here to describe the connection. |
| Shared key | Uncheck Automatically generate a shared key, then paste in the shared key for the connection here, using the key copied from the server instance created previously. |
| Remote network | Enter **`192.168.1.0/24`** here. |

After filling in those fields, click Save. The configuration of the client is complete. No firewall rules are required on the client side because the client only initiates outbound connections. The server never initiates connections to the client.

### Note

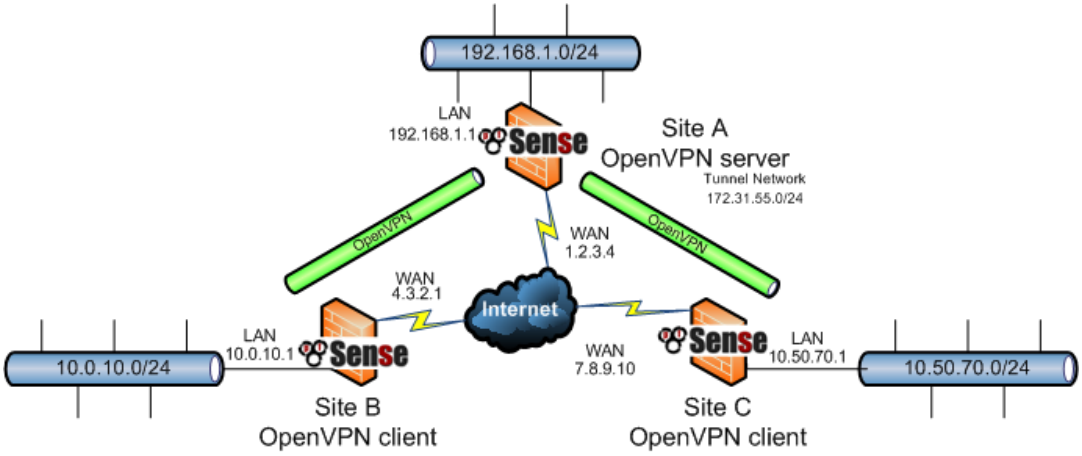
With remote access PKI configurations, frequently you do not define routes and other configuration options in the client configuration, but rather push those options from the server to the client. With shared key deployments, you must define routes and other parameters on both ends as needed (as described previously, and later in the section called “Custom configuration options”), you cannot push from client to server when using shared keys.

## Testing the connection

The configuration is now complete and the connection should 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 the section called “Troubleshooting OpenVPN”.

# Site to Site Example Configuration (SSL/TLS)

**Figure 18.12. OpenVPN example site to site SSL/TLS network**



This section describes the process of configuring a site to site connection using SSL/TLS. This method can be 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 your main location will be the server side and the remote offices will act as clients, though if one location has a static IP 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, as with the remote access OpenVPN configuration, there will be a dedicated subnet in use for the OpenVPN interconnection between networks. The example configuration described here is depicted in Figure 18.12, “OpenVPN example site to site SSL/TLS network”.

**`172.31.55.0/24`** is used as the IPv4 Tunnel Network. The way OpenVPN allocates IPs is the same as for remote access clients, each connecting client gets a /30 subnet to interconnect itself with the server. The following sections describe how to configure the server and client sides of the connection. Any subnet can be used for this so long as it does not overlap any other subnet currently in use on your network.

In order for the server to reach the client networks behind each connection, you need both a **`route`** to the network to tell the operating system that OpenVPN knows about that network, and also an **`iroute`** that tells OpenVPN to which specific connection a given subnet belongs. More detail on this will follow in the example.

# Configuring SSL/TLS Server Side

Before the VPN can be configured, you need to create a CA and Certificate structure for this VPN. First, create a CA unique to this VPN. 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 Chapter 8, *Certificate Management*. For this example, the CA will be called *`S2SCA`*, the Server CN will be *`serverA`*, the clients will be *`clientB`* and *`clientC`*.

Browse to VPN → OpenVPN and click on the Server tab. The following fields are configured, with everything else left at defaults. These options are discussed in detail earlier in the chapter. Use values appropriate for your network, or the defaults if you are unsure.

| | |
|---|---|
| Server Mode | Select **`Peer to Peer (SSL/TLS)`**. |
| Protocol | Select **`UDP`**. |
| Device Mode | Select **`tun`**. |
| Interface | Select **`WAN`**. |
| Local Port | Enter **`1194`** unless you have another active OpenVPN server, in which case you should use a higher port. |
| Description | Enter something here to describe the connection. |
| TLS Authentication | Check this box if you want to also do TLS authentication as well as SSL. This is optional, but adds another layer of security. As with Shared Key mode, after saving you can go back and copy this key, then paste it into the clients later. |
| Peer Certificate Authority | Select the CA created at the beginning of this process. |
| Peer Certificate Revocation List | If you created a CRL, select it here. |
| Server Certificate | Select the server certificate created at the beginning of this process. |

| | |
|---|---|
| Tunnel Network | Enter **172.31.55.0/24** here. |
| Local Network | Enter **192.168.1.0/24** here. |
| Advanced Options | In this box, you will need to add a route for each client subnet that will be reachable via this VPN. You will also likely want to push routes for those same networks to ensure that your remote sites can reach each other. Following the example diagram above, this would look like:<br>`route 10.0.10.0 255.255.255.0;`<br>`route 10.50.70.0 255.255.255.0;`<br>`push "route 10.0.10.0 255.255.255.0";`<br>`push "route 10.50.70.0 255.255.255.0";` |

If there are more networks on the server side that should be reachable via the clients, such as networks reachable via static routes, other VPNs, etc, you may add them as additional pushed routes.

That is everything that must be configured on this screen for OpenVPN. Click Save.

If you chose to use TLS Authentication, You will need to copy the TLS key which was just generated for use on the client system. From the list of OpenVPN server instances, click next to the one that was just created. Find the TLS Authentication box and select all of the text inside, then copy the text to the clipboard. You can either save this to a file, or paste it into a text editor such as Notepad temporarily.

Next you will need to add a firewall rule on WAN allowing access to the OpenVPN server. Specify the protocol **UDP**, source IP as the client's IP address if it has a static IP, or **any** if its IP is dynamic. Destination is the **WAN Address**, and destination port is **1194** in this instance. Figure 18.11, “OpenVPN example site to site WAN firewall rule” shows the firewall rule used for this example.

Apply changes after the firewall rule is added.

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 that site's certificate so that it may be properly routed. Under VPN → OpenVPN, click on the Client Specific Overrides tab, and click to add a new override. On this screen, fill in the fields as follows:

| | |
|---|---|
| Common Name | Enter the CN of the first client site. In our example, that is *clientB*. |
| Advanced | Enter an **iroute** statement for the first client site's subnet. In our example, that is *clientB*'s subnet, **10.0.10.0**, it will look like:<br>`iroute 10.0.10.0 255.255.255.0;` |

Add another override for the second site, adjusting the CN and **iroute** statements as needed.

Next you will want to export the certificates and keys you need. These can be obtained by going to System → Cert Manager and clicking the links to export the following items:

- CA Certificate
- Client site certificate (`.crt`) for each client location.
- Client site key (`.key`) for each client location.

You *do not* need to export the CA key, server certificate, or server key.

That completes the server setup, next, configure the clients.

# Configuring SSL/TLS Client Side

First, on the client, you will need to import the CA certificate along with that site's certificate and key. This is the same CA and 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 Chapter 8, *Certificate Management*.

After the certificates have been imported, browse to VPN → OpenVPN and click on the Client tab. The following fields are configured, 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. In our example this is **`1.2.3.4`**. |
| Server Port | Enter **`1194`** or whichever port was configured on the server. |
| Description | Enter something here to describe the connection. |
| TLS Authentication | Check Enable authentication of TLS packets if you chose on the server to also do TLS authentication as well as SSL. This is optional, but adds another layer of security. 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. |

After filling in those fields, click Save. The configuration of the client is complete. No firewall rules are required on the client's WAN because the client only initiates outbound connections. The server never initiates connections to the client.

### Note

With remote access PKI configurations, frequently you do not define routes and other configuration options in the client configuration, but rather push those options from the server to the client. If you have more networks to reach on the server side, they should be pushed from there.

## Testing the connection

The configuration is now complete and the connection should 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 the section called “Troubleshooting OpenVPN”.

# Checking the Status of OpenVPN Clients and Servers

Added in pfSense 2.0, 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 instance on the status page.

For OpenVPN servers in SSL/TLS server mode, the status will provide a list of connected remote clients, along with their usernames or certificate common names, as seen in Figure 18.13, “OpenVPN

Status for SSL/TLS server with one connected client". You may also disconnect clients from this screen by clicking the at the end of the client row. For these servers a Show Routing Table button will also be displayed. Clicking this button will show a table of networks and IPs connected through each client certificate.

### Figure 18.13. OpenVPN Status for SSL/TLS server with one connected client

Remote Access UDP:1192 Client connections

| Common Name | Real Address | Virtual Address | Connected Since | Bytes Sent | Bytes Received |
|---|---|---|---|---|---|
| jim | 192.168.20.22:1194 | 192.168.138.2 | Sat Jul 13 19:19:17 2013 | 22481 | 19792 |

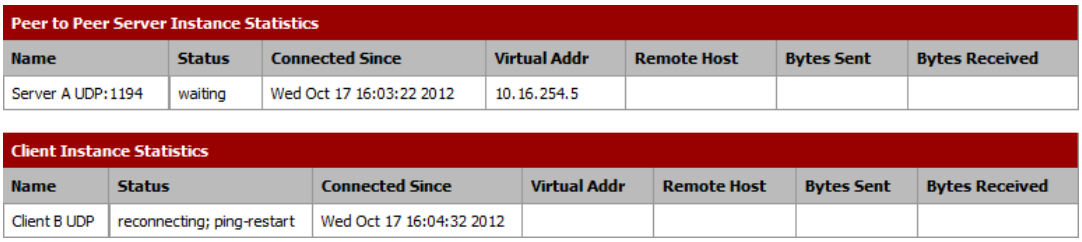
Running

Show Routing Table - Display OpenVPN's internal routing table for this server.

For OpenVPN servers in shared key mode, the status will indicate whether it's running and waiting on connections, or if the remote client has connected.

For OpenVPN clients, the status indicates whether the a connection is pending or active.

### Figure 18.14. OpenVPN Status showing a server waiting for a connection, and a client attempting to reconnect



Peer to Peer Server Instance Statistics

| Name | Status | Connected Since | Virtual Addr | Remote Host | Bytes Sent | Bytes Received |
|---|---|---|---|---|---|---|
| Server A UDP:1194 | waiting | Wed Oct 17 16:03:22 2012 | 10.16.254.5 | | | |

Client Instance Statistics

| Name | Status | Connected Since | Virtual Addr | Remote Host | Bytes Sent | Bytes Received |
|---|---|---|---|---|---|---|
| Client B UDP | reconnecting; ping-restart | Wed Oct 17 16:04:32 2012 | | | | |

# Permitting traffic to the OpenVPN server

After setting up an OpenVPN server, a firewall rule to permit traffic to the OpenVPN server is required. Browse to Firewall → Rules, and to the WAN tab, then click . For the example configuration here, protocol **UDP** will be chosen, with **any** source, destination **WAN Address**, and destination port **1194**. This rule is depicted in Figure 18.15, "OpenVPN server WAN rule".

### Figure 18.15. OpenVPN server WAN rule

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | UDP | 4.3.2.1 | * | 1.2.3.4 | 1194 (OpenVPN) | * | | Allow site B OpenVPN |

If you know which source addresses your clients will be connecting from, you can specify a source network or alias rather than leaving the server open to the entire Internet. This is usually impossible where you have roaming clients. There is not much risk to leaving this open, however, as with certificate based authentication you have lesser 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. The exception to this is when a firewall was upgraded from pfSense 1.2.3 where OpenVPN filtering was not possible. A rule to allow all traffic over OpenVPN is added during the upgrade from 1.2.3 to 2.x in order to preserve the firewall's previous behavior after the upgrade. To allow traffic from OpenVPN clients to make connections to resources on the server side, you must add firewall rules under Firewall → Rules, on the

OpenVPN tab. As with other aspects of the firewall, these rules will only match traffic coming into the system from the *client* side, not traffic leaving from the server side, so craft your rules appropriately. If you need to reach devices on the client side, add rules on the OpenVPN tab on the client firewall as well.

# OpenVPN clients and Internet Access

If you simply want to NAT your OpenVPN clients to your WAN IP so they can access the Internet using the OpenVPN connection, rules should automatically allow this. If you want to have more fine grained control, you need to enable Advanced Outbound NAT and edit the Outbound NAT rule for your IPv4 Tunnel Network subnet(s), which should automatically appear after switching form Automatic to Manual. See the section called “Outbound NAT” for more details on Outbound NAT.

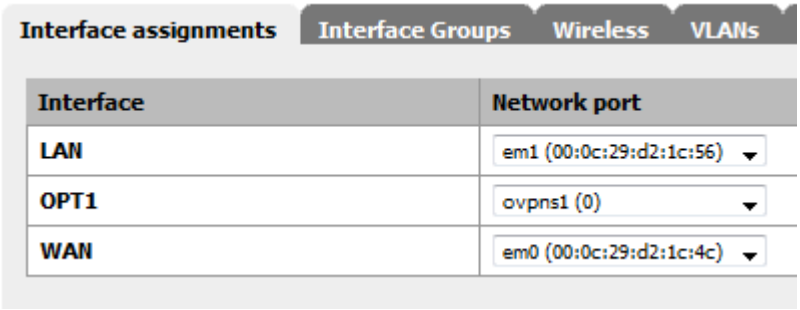
# NAT with OpenVPN Connections

In order to do complex NAT, policy routing, or tunnel-specific filtering, you must assign the OpenVPN interface to an OPT interface and configure it accordingly. This section describes how to accomplish NAT for OpenVPN clients.

## Interface assignment and configuration

Browse to Interfaces → Assign and assign the appropriate `ovpns` or `ovpnc` interface as an OPT interface. The name of the OpenVPN device will depend on how it was configured. Server instances are `ovpns`*x*, clients are `ovpnc`*x*. You will have one interface per OpenVPN server and client configured on the system. Figure 18.16, “Assign OpenVPN interface” shows `ovpns1` assigned as OPT1.

**Figure 18.16. Assign OpenVPN interface**




Now browse to the interface page for the previously assigned interface, Interfaces → OPT1 for the example shown in Figure 18.16, “Assign OpenVPN interface”. First check the Enable interface box at the top of the page, and enter an appropriate description in the Description field. Select **none** in the Type box. This will not configure any IP information on the interface, which is necessary since OpenVPN itself must configure these settings on the `ovpns1` interface. Click Save to apply these changes. This does nothing to change the functionality of OpenVPN, it simply makes the interface available for firewall rule and NAT purposes.

## Filtering with OpenVPN

Now that you have the OpenVPN interface assigned, browse to Firewall → Rules and click the tab for the interface you just assigned. Here you can add firewall rules just like any other interface that will apply to traffic initiated by OpenVPN clients. For more information on firewall rules, refer to Chapter 10, *Firewall*. The rules on the OpenVPN tab will still apply to traffic on an assigned OpenVPN interface. The OpenVPN tab rules are considered first, and then the assigned interface rules. To ensure that you do not allow more traffic than desired, make sure that none of the rules on the OpenVPN tab would also match rules on the assigned interface tab.

# Policy Routing with OpenVPN

When you have the OpenVPN interface assigned and enabled, you will also get an automatic gateway entry under System → Routing, on the Gateways tab. With this, you can direct traffic into the VPN using the Gateway field on your LAN or other internal interface firewall rules. See the section called "Policy routing" for more information on policy routing. If you use this to direct traffic across a VPN, you will need to either do NAT on the VPN interface before it leaves (for VPN services such as StrongVPN and similar) or have the NAT done on the other side before it reaches the actual Internet connection.

### Note

Do not use this automatic gateway for static routes. Use the **Remote Network** field in the VPN configuration or **route** statements in the advanced options, but defining a static route using the automatic OpenVPN gateway will not work effectively.

# NAT with OpenVPN

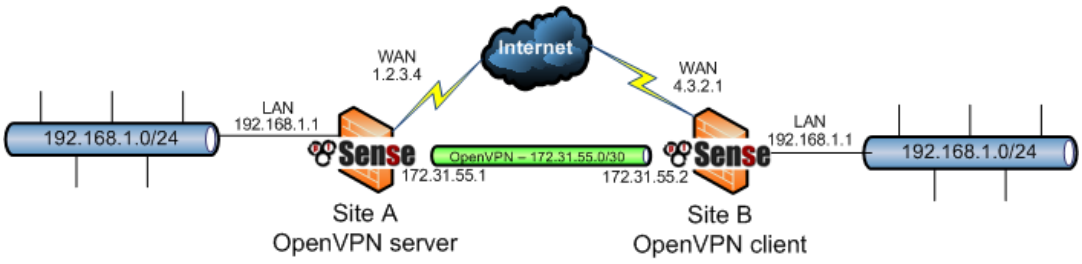
With the OpenVPN interface assigned, NAT rules can also be applied the same as with any other interface. This is useful when you must connect two conflicting subnets. If you have two networks using a 192.168.1.0/24 LAN subnet that you need to connect using a site to site VPN, they cannot communicate across VPN without NAT. Hosts on a 192.168.1.0/24 subnet will never reach the other end of the VPN to communicate with the remote 192.168.1.0/24 subnet, because that network is always treated as local. However with NAT, you can make the remote side function as if it were using a different IP subnet.

### Note

This will work fine for many protocols, but for some that are commonly desirable across VPN connections, primarily SMB/CIFS file sharing between Windows hosts, will not function in combination with NAT. If you are using a protocol that is not capable of functioning with NAT, this is not a viable solution.

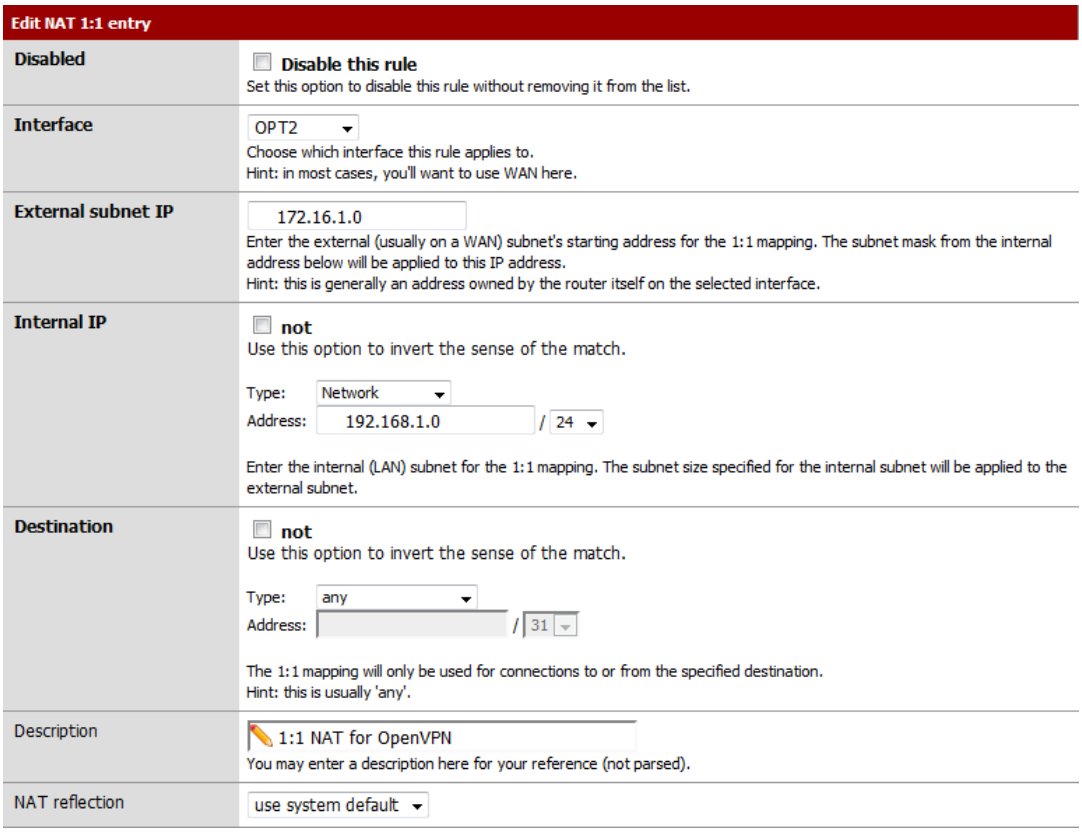
Figure 18.17, "Site to site with conflicting subnets" shows an example where both ends are using the same subnet. After assigning the `tun` interface to an OPT interface on both sides, as described in the section called "Interface assignment and configuration", 1:1 NAT can be applied.

**Figure 18.17. Site to site with conflicting subnets**

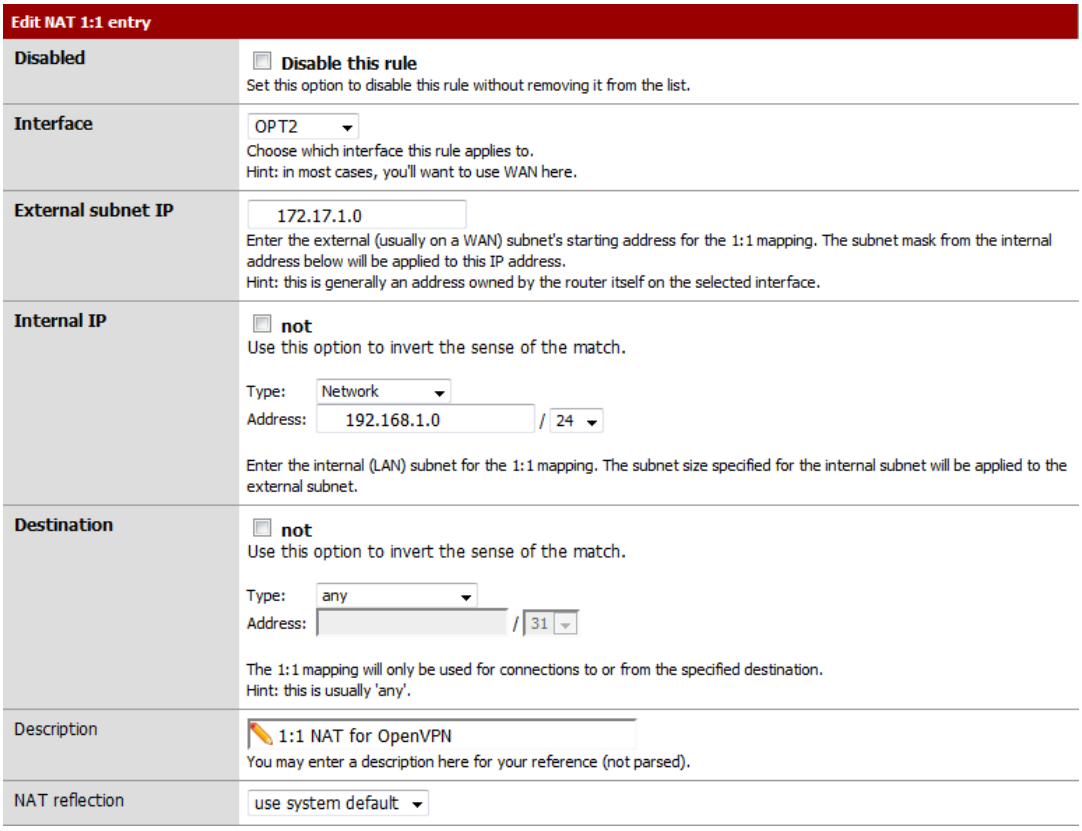



The traffic from Site A will be translated to 172.16.1.0/24, and Site B will be translated to 172.17.1.0/24. A 1:1 NAT entry will be added on each end to translate the entire /24 range. To reach Site A from Site B, 172.16.1.x IP addresses will be used. The last octet in the 192.168.1.x IP will be translated to the last octet in the 172.16.1.x translated IP, so to reach 192.168.1.10 at Site A from Site B, you would use 172.16.1.10 instead. To reach 192.168.1.50 at Site B from Site A, you would use 172.17.1.50 instead. Figure 18.18, "Site A 1:1 NAT configuration" and Figure 18.19, "Site B 1:1 NAT configuration" show the 1:1 NAT configuration for each side, where the `tun` interface is assigned as OPT2.

## Figure 18.18. Site A 1:1 NAT configuration




## Figure 18.19. Site B 1:1 NAT configuration



In the OpenVPN configuration on both sides, the Remote network must be specified as the translated IP subnet, not as 192.168.1.0/24. 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 some circumstances. This section covers multi-WAN considerations with OpenVPN server and client configurations.

# OpenVPN assigned to a Gateway Group

Starting with pfSense 2.1, you can now select a Gateway Group (the section called “Gateway Groups”) as the Interface for an OpenVPN instance. Such a gateway group must only have one gateway per tier. When creating the gateway group, you may select a VIP to use for that tier also. When selected for a VPN server, the gateway group's tier 1 interface or VIP will be used for binding first. If the gateway goes down, it will move to tier 2, etc. If the tier 1 gateway comes back up, the VPN will resume operating on that WAN. 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 certain circumstances, such as needing both WANs usable concurrently with the VPN. When used with OpenVPN clients, the outbound interface will be switched according to the gateway group's 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 your configuration.

## OpenVPN server using TCP

While TCP is generally not the preferred protocol for OpenVPN, as described earlier in this chapter, 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. You need a firewall rule on each WAN interface.

## OpenVPN server using UDP

OpenVPN servers with UDP are also multi-WAN capable, but with some caveats that aren't applicable with TCP, because of the way pf's multi-WAN routing functions. In some cases, each WAN must have its own OpenVPN server. You can use the same certificates for all the servers. Only two parts of the OpenVPN configuration must change.

### Multiple Server Method

#### 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 the Interface WAN interface used by that server.

### Port forward method

A somewhat easier option can be to bind the OpenVPN server to the LAN interface, or **`Localhost`**, and then use a port forward from each WAN to direct the OpenVPN port to the IP upon which the service is listening. Using this method, pf's reply-to functionality will ensure that the return traffic flows back to the proper source via the proper interface. This method requires manual intervention when used with the client export package, however. You must specify the WAN IP(s) when exporting, as the default exporter settings would leave it attempting to connect to the server's LAN IP over the Internet, which isn't routable.

### 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 your OpenVPN servers are running on IPs 1.2.3.4 and 4.3.2.1, both using port 1194, the **`remote`** lines in your client configuration file will be as follows.

```
remote 1.2.3.4 1194 udp
remote 4.3.2.1 1194 udp
```

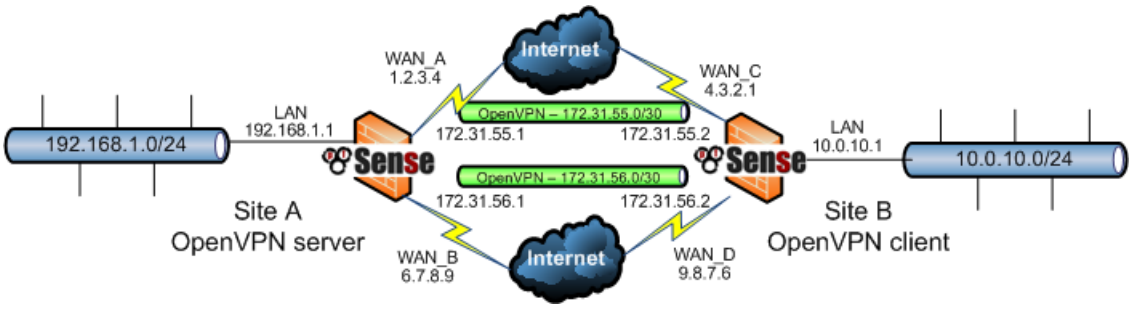
For clients configured on pfSense, the first **`remote`** is configured by the options given in the GUI. The second **`remote`** is specified in the custom options field.

## OpenVPN Clients and Multi-WAN

OpenVPN clients configured on the firewall will respect the Interface chosen when configuring the client, and a static route is added automatically behind the scenes to allow it to function as desired. If for some reason the interface must be set to any, the client will follow the system routing table when making the connection to the OpenVPN server. This was the default behavior in pfSense 1.2.3 and earlier. To use an OPT WAN interface, select the Interface as required. If you must set the client to an Interface value of any, then you will need to enter a static route to direct traffic to the remote endpoint of the OpenVPN connection.

## OpenVPN Site-to-Site with Multi-WAN and OSPF

**Figure 18.20. Example OpenVPN setup involving OSPF across multiple WANs**




Building upon concepts from earlier in the chapter, it's possible to setup a redundant VPN setup using a dynamic routing protocol like OSPF, like seen in Figure 18.20, “Example OpenVPN setup involving OSPF across multiple WANs”.

First, setup instances on each WAN for the remote sites. These should be shared key site-to-site tunnels, with no Remote Networks filled in, only Tunnel Network addresses.

- On the server side setup two servers, 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`*)

- On the client side setup two clients, each paired up with one of the above servers, matching the IP addresses and port numbers involved.
- Ensure these 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.

Make sure these OpenVPN connections link up between client and server. You should be able to ping the tunnel address on both sides. If the tunnels do not establish, see the section called "Troubleshooting OpenVPN" for suggestions on troubleshooting the connection. Ensure in your OpenVPN firewall rules that you 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 you can use 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 and the traffic cannot leave that segment.

Once both instances are connected, you can move on to setting up OSPF.

First, on both firewalls you must install the Quagga-OSPF package from System → Packages on the Available Packages tab. Once installed, go to Services → Quagga OSPFd, this is where the OSPF setup is configured.

On the Interfaces tab, add each OpenVPN interface. Set the cost to *`10`* on the primary link and *`20`* on the secondary, etc. Then add your LAN and other internal interfaces as passive interfaces.

Once the interfaces have been added, switch to the Global Settings tab. Set a Master Password. It doesn't really matter what it's set to, it's used internally for accessing the status daemon, but it needs to be set. Set the Router ID to an IP-address-like value, by that we mean a value that looks like an IP address, e.g. *`192.168.1.1`*. The Router ID is unique on each device, which is why setting it to the router's LAN IP address is a good practice. Lastly, 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 you may use whatever value you like. The Area ID is the same for all routers involved in this VPN setup. Press Save, and the OSPF setup on that router is complete. Once OSPF has been configured on all routers, they should be pairing up properly.

After OSPF has been setup on both ends, the Status tab should show a full peering with each instance on each wan, and you should see the routes obtained via OSPF listed. Once that happens, you can try unplugging/replugging WANs and refreshing the status (will a ping going between internal networks) to test the connection.

# OpenVPN and CARP

OpenVPN is interoperable with CARP. To provide a high availability OpenVPN solution with CARP, configure your clients to connect to a CARP VIP, and configure the OpenVPN server to use the CARP IP with the Interface option. In pfSense 1.2.x, the OpenVPN configuration cannot be synchronized with your secondary firewall, so you must manually enter it into both firewalls. On pfSense 2.x, settings 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. CARP is discussed further in Chapter 25, *Firewall Redundancy / High Availability*. As of pfSense 2.0.2, the firewall will automatically shut down OpenVPN instances as needed when a CARP node is in a backup state. This prevents OpenVPN from making unnecessary outbound connections in client mode, and in both client and server mode it prevents OpenVPN from maintaining unnecessary routes. When the CARP 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 usually the preferable way of connecting VPN clients, but OpenVPN also offers the option of using `tap` interfaces and bridging clients directly onto your LAN or other internal network. This can make the remote clients appear to be on your local LAN. Most of the settings for setting up 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

The first step in setting up such a bridge is to select **tap** from the server's Device Mode drop-down.

# Tunnel Network

You will also want to make sure that the IPv4 Tunnel Network and IPv6 Tunnel Network boxes are empty. The way that 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

If Bridge DHCP is selected, DHCP will be passed through to the bridged interface that will be setup later. In the most common scenario, this would be LAN. Using this method, connecting clients would receive IPs from the same DHCP pool used by directly wired LAN clients.

# Bridge Interface

The Bridge Interface drop-down does not actually create the bridge, it only indicates to OpenVPN which interface will be used for the bridge. In most cases, this would be LAN. This setting 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, you may optionally supply a DHCP range 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 LAN, and the interface setting above will be ignored. This allows you to set aside a range of IPs for use only by OpenVPN clients, so they may be contained within a portion of your internal network, rather than consuming IPs 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 interface must be assigned and bridged to your 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 the section called “Interface assignment and configuration”.

## Create Bridge

Once the VPN interface has been assigned, navigate to Interfaces → (assign) on the Bridges tab. From there, click to create a bridge. On the resulting screen **ctrl**-click both the VPN interface and the interface to which you want it to be bridged (e.g. **LAN**), then click Save. More information on bridging can be found in Chapter 13, *Bridging*.

# Connect with Clients

Clients connecting to the VPN must also be set to use `tap` mode. Once that has been set, you can connect with a client (such as one exported using the OpenVPN Client Export package) and the clients

should receive an IP inside of your internal subnet, as if they were on your LAN. They will receive broadcast and multicast traffic as well.

# Custom configuration options

OpenVPN offers dozens of configuration options, many beyond the most commonly used fields that are presented in the GUI. This is why the custom configuration options box exists. You can fill in an unlimited number of additional configuration options, separated by semicolons. This section covers the most frequently used custom options individually. There are many more, though rarely needed. The OpenVPN man page [http://openvpn.net/index.php/open-source/documentation/manuals/65-openvpn-20x-manpage.html] details them all. 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. You can 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 `Options error: Unrecognized option or missing parameter(s)` followed by the option that caused the error.

# Routing options

To add additional routes for a particular OpenVPN client or server, you use the **`route`** custom configuration option. 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 an OpenVPN server configuration using PKI, you can also push additional routes to clients. To push the routes 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 you to change the default gateway of the client to the OpenVPN connection, so all the traffic from the client is pushed across 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. If you wish to do this manually, add the following custom option:

```
push "redirect-gateway def1"
```

You can also enter this as a custom option on the client side by using **`redirect-gateway def1`** without specifying **`push`**. (Note the option is the letters "**`def`**" followed by the digit *one*, not the letter "L".)

# Specifying IP address to use

The **`local`** custom option allows you to specify the IP address the OpenVPN service will use. This is no longer required as you can select an interface or VIP from the Interface drop-down to perform this task. If you need to do this manually, it can be either an IP address, such as **`local 1.2.3.4`**, or a FQDN such as:

```
local myopenvpn.dyndns.org
```

This is mostly used in multi-WAN scenarios, as described in the section called “OpenVPN and Multi-WAN”, or in combination with CARP VIPs.

# Sharing a Port with OpenVPN and a Web Server

If you want to be extra sneaky/careful with your OpenVPN server, you can take advantage of OpenVPN's **`port-share`** capability that allows it to pass any non-OpenVPN traffic to another IP behind the firewall. The usual use case for this would be to run your OpenVPN server on port tcp/443, and in place of a port forward, let OpenVPN hand off the HTTPS traffic to a web server.

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 you 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 (or whatever IP used for OpenVPN) on port 443. You do not need any port forwards or firewall rules 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 you point an OpenVPN client there, it should connect and work fine, and if you point a web browser at the same IP, you should 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. These follow the Cisco avpair standard. The values you can specify through these avpair values are:

| | |
|---|---|
| inacl | Inbound firewall rules to govern traffic from the client to the server. |
| outacl | Outbound firewall rules to govern traffic from the server to the client. |
| dns-servers | DNS servers to push to the client. |
| route | Additional route statements to push to the client. |

# Troubleshooting OpenVPN

Should you encounter any problems trying to use OpenVPN, this section provides information on troubleshooting the most common issues users encounter.

## Check OpenVPN Status

The first place to look is Status → OpenVPN. The connection status for each VPN will be shown there. If a VPN is connected, waiting, reconnecting, etc, it would be indicated on that screen. For more information, see the section called “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 you see traffic for the tunnel itself being blocked, such as traffic to the WAN IP on port 1194, then adjust your WAN firewall rules accordingly. If you see traffic 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 you can do about that other than getting the software on the device fixed. You can verify this by running **tcpdump** on the inside interface of the firewall connected to the network containing the device. Troubleshooting with **tcpdump** is covered in the section called "Using tcpdump from the command line". If you see traffic going out the inside interface on the firewall, but no replies coming back, the device is not properly routing its reply traffic (or could potentially be blocking it via a 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. |
| 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 something similar to the following (the number following `openvpn` will differ, it is the process ID of the OpenVPN process making the connection).

```
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
```

If you do not see the `link remote` and `Peer Connection Initialized` messages upon 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 you have recently disabled or removed an IPsec tunnel, 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 added. For site to site VPNs, you should see routes for the remote network(s) to the appropriate `tun` or `tap` interface. If the routes are missing or incorrect, your Local Network, Remote Network, or custom options are not configured correctly. If you are using a shared key setup and not PKI, ensure that you are not using "push" commands are instead adding routes to both ends using "route" custom options, as in the section called “Routing options”.

# Test from different vantage points

If the connection shows as being up in the logs, but doesn't work from your LAN, try it from the firewall itself, first using the inside interface being used for the OpenVPN connection (typically LAN) as the ping source. If that doesn't work, SSH into the firewall and choose option 8 for a command prompt. Run **`ping x.x.x.x`** at the command line, replacing *`x.x.x.x`* with an IP on the remote side of the VPN. This will cause the traffic to be initiated from the IP of the `tun` interface being used by OpenVPN. This can help narrow down routing problems on the remote network.

# Trace the traffic with tcpdump

Using **tcpdump** to determine where the traffic is seen and where it isn't 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 Chapter 30, *Packet Capturing*.

# Routes will not push to a client

If you are attempting to use the Local Network box 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 you're using an SSL/TLS server setup with a Tunnel Network larger than a **`/30`**. OpenVPN's **`server`** mode only takes effect if you're using a subnet large enough to contain multiple clients, such as a **`/24`**.
- If the client is running on Windows Vista, Windows 7, or similar, try running the client as Administrator. Some versions of the OpenVPN client require Administrator mode to apply routes to the system's routing table.
- If you're using a shared key setup, pushing routes will not work. Use 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, 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 of the server.

According to the *OpenVPN FAQ [http://www.openvpn.net/index.php/documentation/faq.html]*, 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 your server you see something like this in the ifconfig output:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
 inet6 fe80::202:b3ff:fe03:8028%tun0 prefixlen 64 scopeid 0xc
 inet 192.168.100.1 --> 192.168.100.2 netmask 0xffffffff
 Opened by PID 27841
```

While the client shows:

```
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> metric 0 mtu 1500
        inet6 fe80::202:b3ff:fe24:978c%tun0 prefixlen 64 scopeid 0xa
        inet 192.168.100.6 --> 192.168.100.5 netmask 0xffffffff
        Opened by PID 1949
```

In this case, you likely cannot ping .5 or .1. .5 because it's a virtual address, and .1 because you have 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 your default route instead.

There are many cases where you can ping the far side of an OpenVPN tunnel, but not the local. This is also counter-intuitive, but works especially in cases where you have a site-to-site link. If the server shows its tun addresses as "*`x.x.x.1`* -> *`x.x.x.2`*" and the client shows the reverse - "*`x.x.x.2`* -> *`x.x.x.1`*", then you can ping the far side from both ends.

# Cannot route to clients on an SSL/TLS site-to-site tunnel even though all the settings appear correct

If you setup an SSL/TLS site-to-site tunnel, and all of your routes appear correct, but you still cannot get traffic to flow, check your 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 the section called “IPv4/IPv6 Tunnel Network” for more information. If you need to connect multiple sites to a single server instance, check your setup against the section called “Site to Site Example Configuration (SSL/TLS)”, especially the client-specific overrides and **iroutes**.

# Client Specific Override iroute entry seems to have no effect

If you are trying to setup a site-to-site PKI OpenVPN setup, you need to add an **`iroute`** statement for the client subnet on the Client Specific Overrides tab set for the client certificate's common name.

First, ensure that the common name matches and that the internal route is being learned/added as it should be. You may need to increase OpenVPN's log verbosity (i.e. **`verb 10`** in the custom options) to see if this is working.

Also, for each network you want to use an **`iroute`** statement for, you also need a route statement in the server definition's custom options. The **`route`** statements are for the OS to know that they should be routed to OpenVPN from everywhere else. The **`iroute`** statements are internal to OpenVPN, so it knows which network goes to which client.

# Why do my OpenVPN clients all get the same IP?

If you use the same certificate for all of your clients, which is strongly discouraged, then you may find that 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

If you are importing an existing OpenVPN setup into pfSense 2.0, you may be asking "Where is the DH parameters field?" Well, it's all taken care of behind the scenes.

DH parameters are not specific to a given setup in the way that your certificates or keys are. The GUI in 1.2.3 and below was not smart enough to generate them for you automatically.

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.

# Chapter 19. PPTP VPN

pfSense can act as a PPTP VPN server as one of its VPN options. This is an attractive option because the client is built into every Windows and OS X version released in the past decade. It can also provide passthrough services to an internal PPTP server.

For general discussion of the various types of VPNs available in pfSense and their pros and cons, see Chapter 16, *Virtual Private Networks*.

## PPTP Security Warning

Despite the attraction of its convenience, PPTP *should not be used* under any circumstances because it is no longer secure. This is not specific to the implementation of PPTP in pfSense, any system that handles PPTP is no longer secure. The reason for the insecurity is because PPTP relies upon MS-CHAPv2 which has been completely compromised. If you continue to use PPTP be aware that intercepted traffic can be decrypted by a third party 100% of the time, so it should be considered unencrypted. We strongly advise migrating 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/.

If you have not already, you should read the section called “Cryptographically secure” about VPN security. PPTP is widely used, but it is not a secure VPN solution.

## PPTP and Firewall Rules

By default, when you have PPTP redirection or the PPTP server enabled, hidden firewall rules will be automatically added to WAN to permit TCP 1723 and GRE traffic from any source to the destination address. You can disable this behavior on pfSense 1.2.3 and later releases by checking the Disable all auto-added VPN rules box under System → Advanced. You may wish to do this if you know your PPTP clients will be connecting only from particular remote networks. This prevents potential abuse from arbitrary Internet hosts, though in deployments where users are mobile and will be connecting from numerous locations, it's impossible to know all the subnets users will be coming from so tightening the ruleset is impractical and will cause difficulties for your users.

## PPTP and Multi-WAN

Unfortunately because of the way PPTP works, and the way PF works with the GRE protocol, it is only possible to run a PPTP server on the WAN interface that has your default gateway.

## PPTP Limitations

The state tracking code in the underlying PF firewall software for the GRE protocol can only track a single session per public IP per external server. This means if you use PPTP VPN connections, only one internal machine can connect simultaneously to 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. The only available work around is to use multiple public IPs on your firewall, one per client, or to use multiple public IPs on the external PPTP server. This is not a problem with other types of VPN connections.

This same limitation also means if you enable the PPTP Server or Redirection functionality, no clients NATed to your WAN IP address will be able to connect to any outside PPTP server. The work around to this is to NAT your clients' outbound Internet access to a different public IP address.

Both of these limitations are able to be worked around in most environments, and PPTP is an old, insecure protocol that really should no longer be used. We had planned to address this in a future release, but so far have not come up with a viable solution, and given the total compromise of PPTP as a VPN protocol, it is unlikely that it would be worth any effort spent on attempting to fix the issue.

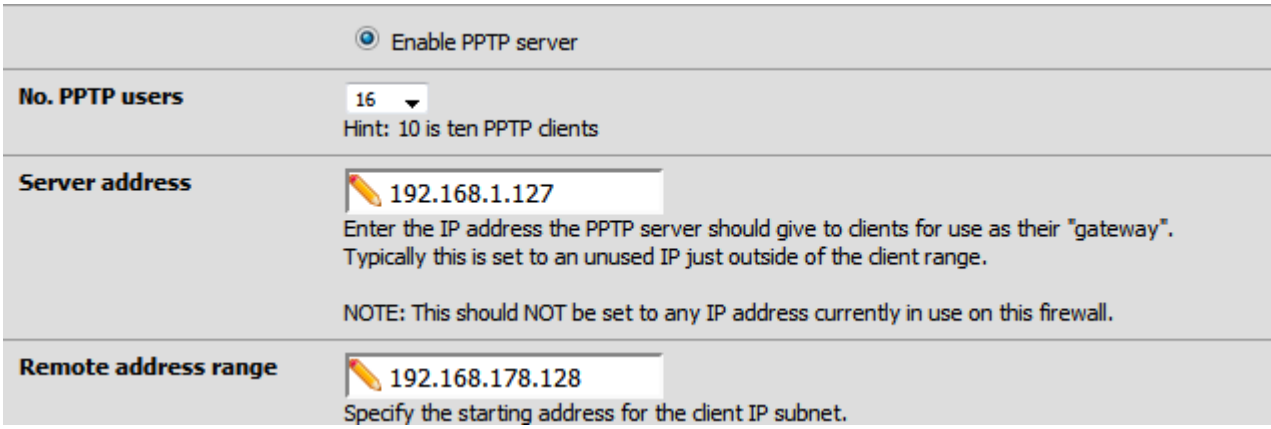
# PPTP Server Configuration

If you still wish to use PPTP, despite the security issues, then to configure the PPTP server, first browse to VPN → PPTP. Select Enable PPTP server.

## IP Addressing

You will need to decide what IP addresses to use for the PPTP server and clients, and now many concurrent clients you want to support. The No. PPTP users field controls how many PPTP users will be allowed to connect at the same time, in this example we selected **16**. The Remote address range is usually a portion of your LAN subnet, such as 192.168.1.128/28 (.128 through .143). These are the addresses to be assigned to clients when they connect. Then select an IP address outside of that range for the Server address, such as 192.168.1.127 as shown in Figure 19.1, “PPTP IP Addressing”.

**Figure 19.1. PPTP IP Addressing**




### Note

This subnet does not have to be contained within an existing subnet on your router. You may use a completely different set of IP addresses if desired.

## Authentication

You can authenticate users from the local user database, or via RADIUS. RADIUS allows you to connect to another server on your network to provide authentication. This can be used to authenticate PPTP users from Microsoft Active Directory (see the section called “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. You can also add a second RADIUS server to use in case the first one fails. For authentication using the local user database, leave that box unchecked. You will have to add your users on the Users tab of the VPN → PPTP screen unless using RADIUS. See the section called “Adding Users” below for more details on the built-in authentication system.

## Require 128 bit encryption

You should require 128 bit encryption where possible. Most PPTP clients support 128 bit encryption, so this should be fine in most environments. PPTP is relatively weak at 128 bit, and significantly more

so at 40 and 56 bit. Unless you absolutely must, you should never use anything less than 128 bit with PPTP, and even then keep in mind that PPTP traffic can be decrypted by an attacker if intercepted.

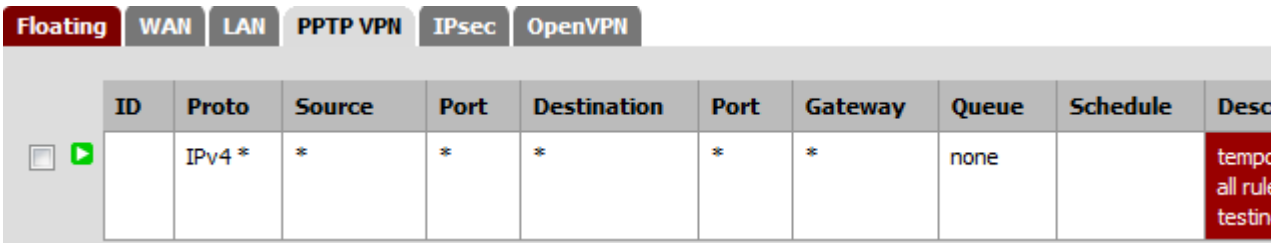
# Save changes to start PPTP server

After filling in the aforementioned items, click Save. This will save your configuration and launch the PPTP server. If you are authenticating your users with the local user database, click the Users tab and enter your users there.

# Configure firewall rules for PPTP clients

Browse to Firewall → Rules and click the PPTP VPN tab. These rules control what traffic is permitted from PPTP clients. Until you add a firewall rule here, all traffic initiated from connected PPTP clients will be blocked. Traffic initiated from your LAN to the PPTP clients is controlled using your LAN firewall rules. Initially you may want to add an allow all rule here for testing purposes as shown in Figure 19.2, “PPTP VPN Firewall Rule”, and once you verify functionality, restrict the ruleset as desired.

**Figure 19.2. PPTP VPN Firewall Rule**




# Adding Users

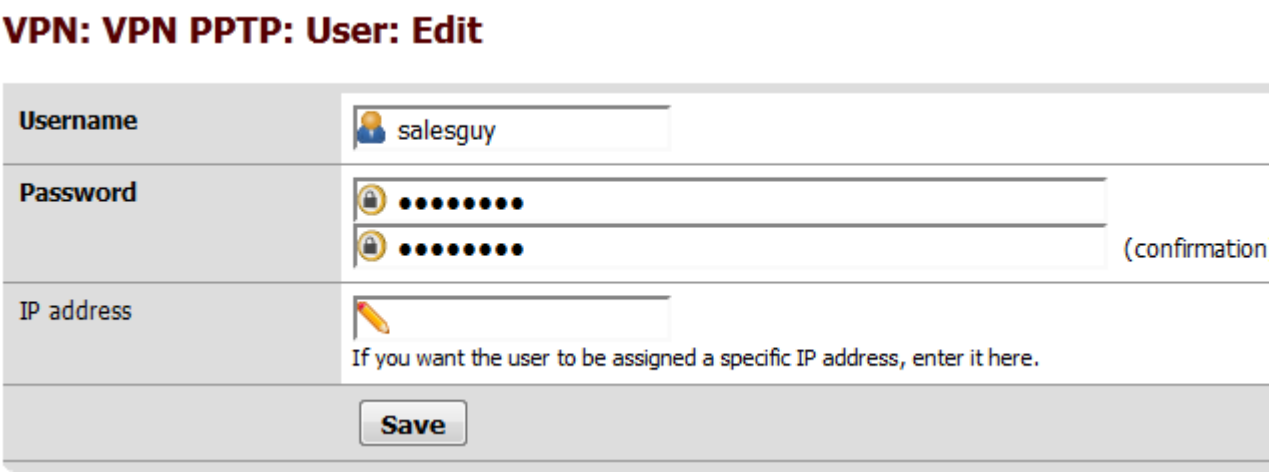
Adding users via RADIUS will vary from one implementation to another. This fact makes it beyond the scope of this section, but should be covered in the documentation for the particular RADIUS server being employed.

Adding users to pfSense's built-in PPTP users system is quite easy. First, click on VPN → PPTP, and then the Users tab. You will be presented with an empty users screen as shown in Figure 19.3, “PPTP Users Tab”. Click the button to add a user.

**Figure 19.3. PPTP Users Tab**

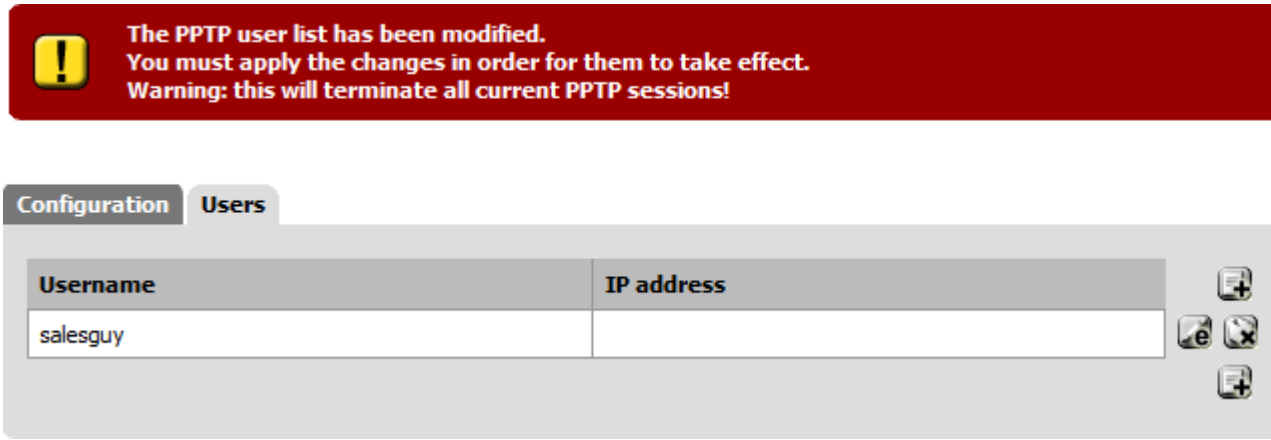

After clicking , the user editing page will appear. Fill it in with the username and password for a user, as in Figure 19.4, “Adding a PPTP User”. You may also enter a static IP assignment if desired.

**Figure 19.4. Adding a PPTP User**



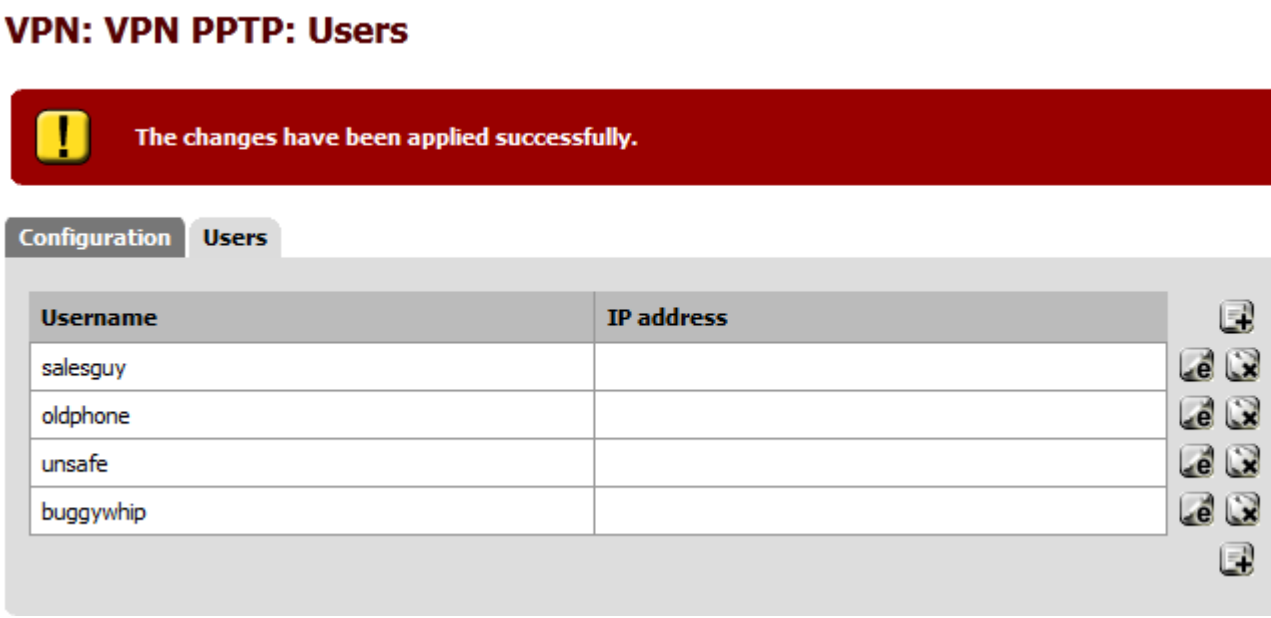

Click Save, and then the user list will return (Figure 19.5, “Applying PPTP Changes”), but before the change will take effect, the Apply Changes button must first be clicked.

**Figure 19.5. Applying PPTP Changes**




Repeat that process for each user you would like to add, eventually you will have a rather full looking user list, as in Figure 19.6, “List of PPTP Users”.

**Figure 19.6. List of PPTP Users**




If you need to edit an existing user, click . Users may be deleted by clicking .

# PPTP Client Configuration

Now that your PPTP server is configured and ready, you will need to configure your PPTP clients. The following sections provide instructions on configuring Windows XP, Windows Vista and Mac OS X for connecting to a PPTP server.

## Windows XP

Open Control Panel, and double click Network Connections (Figure 19.7, “Network Connections”).

**Figure 19.7. Network Connections**


Under Network Tasks, click Create a new connection (Figure 19.8, “Network Tasks”). At the welcome screen of the wizard, click Next.

**Figure 19.8. Network Tasks**

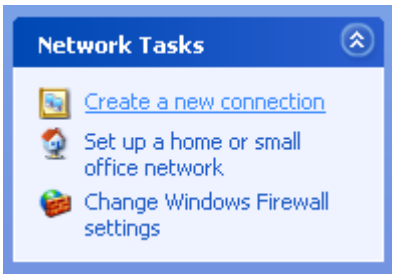
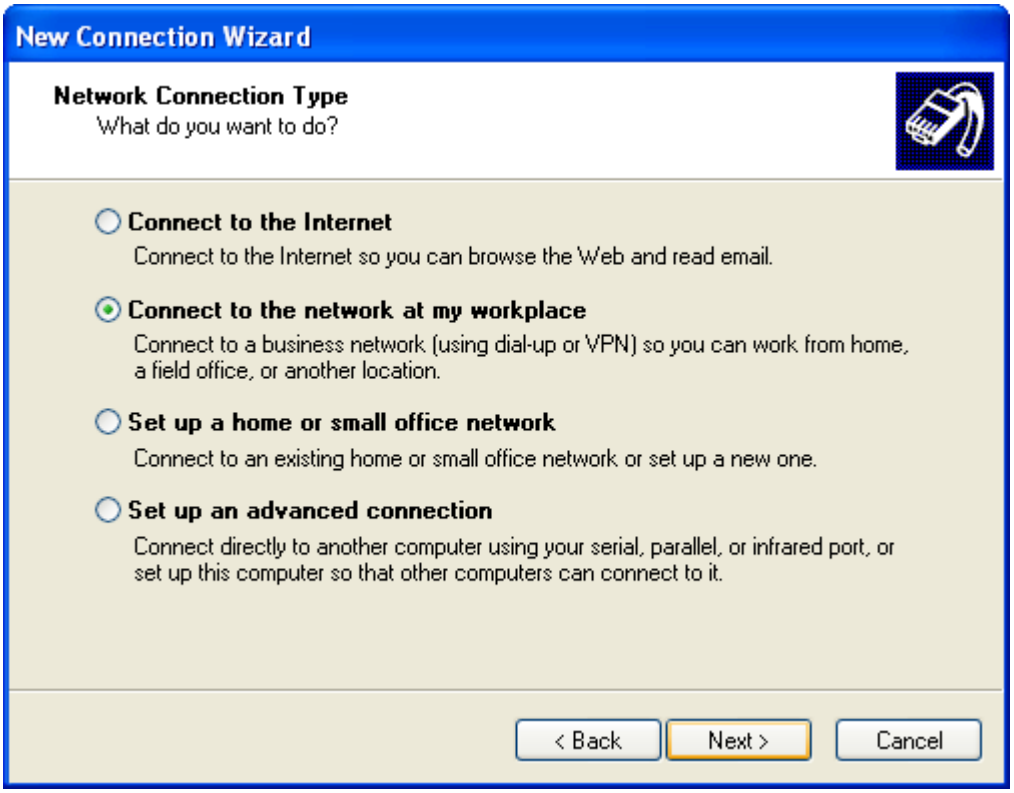


**Figure 19.9. Workplace Connection**



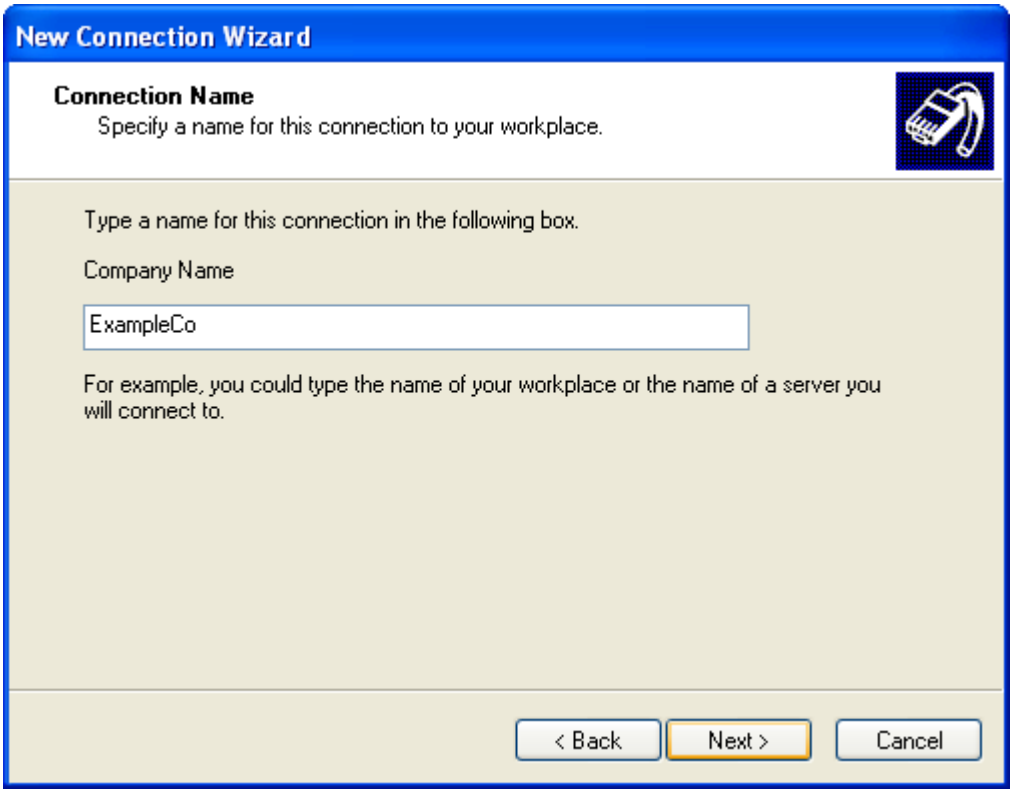

Select Connect to the network at my workplace, as in Figure 19.9, “Workplace Connection”, and click Next.

Select Virtual Private Network connection, like Figure 19.10, “Connect to VPN”, then click Next.

**Figure 19.10. Connect to VPN**


Enter a name for the connection under Company Name, like that in Figure 19.11, “Connection Name”, and click Next.

**Figure 19.11. Connection Name**



Enter the WAN IP of the remote pfSense router under Host Name or IP Address, just like Figure 19.12, “Connection Host”, and click Next, then click Finish (Figure 19.13, “Finishing the Connection”).

**Figure 19.12. Connection Host**

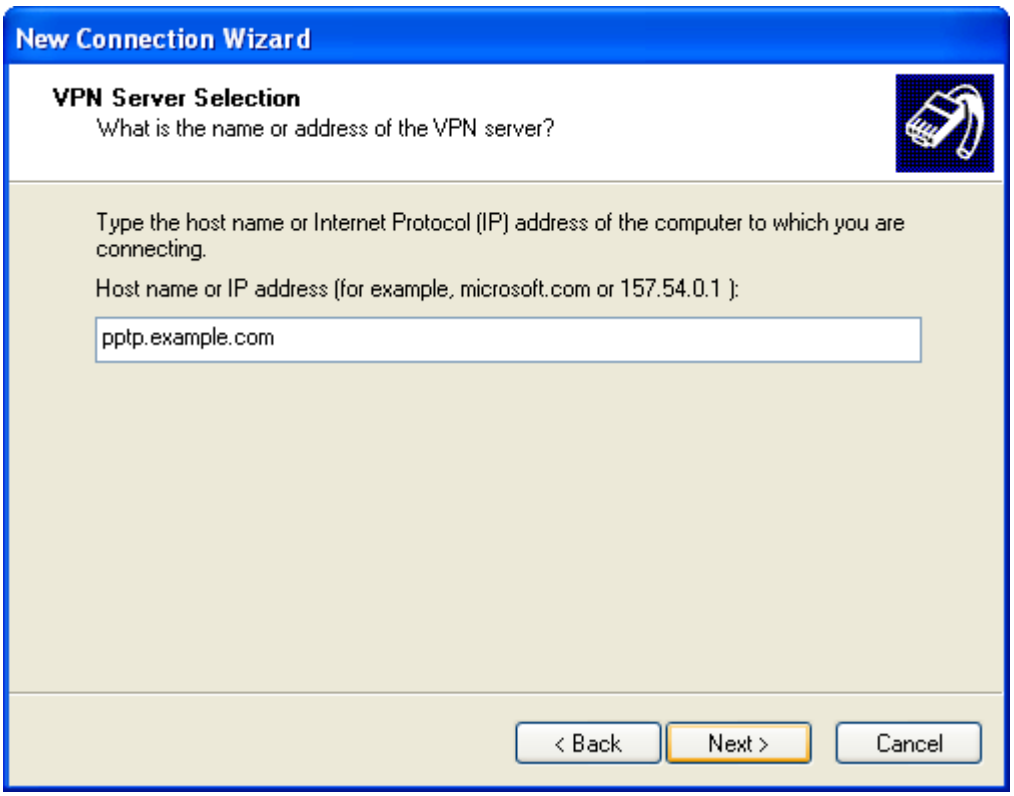
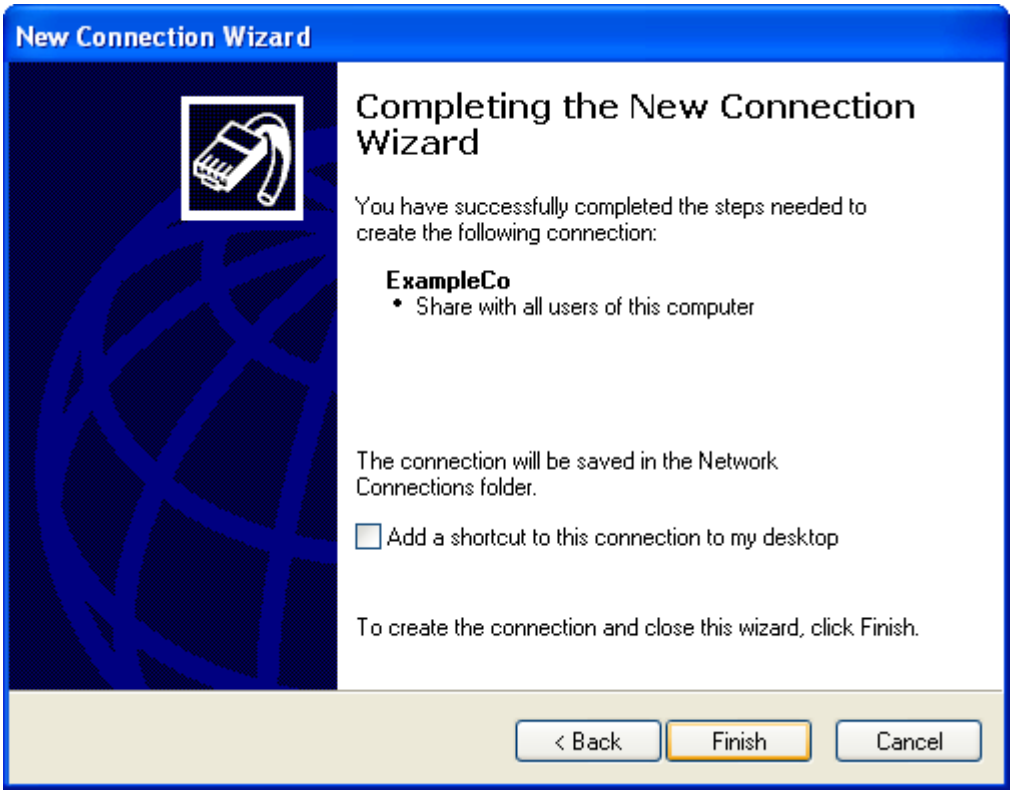


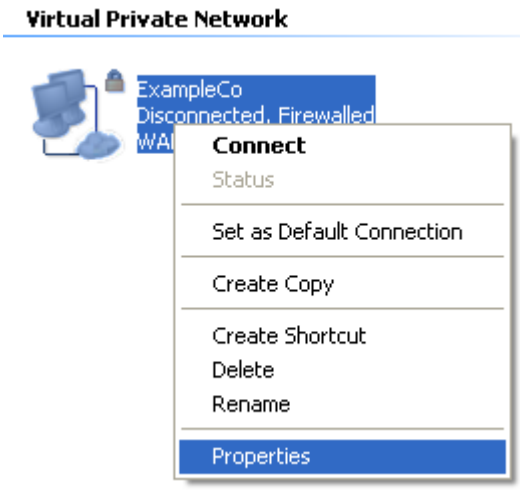

**Figure 19.13. Finishing the Connection**



You now have a PPTP dial-up entry that works like any other dial-up connection. A prompt for the username and password, like that in Figure 19.14, “Connect Dialog”, will show up when the initial connection is attempted. It is best not to connect yet, however. Cancel this dialog if it appears and try again after following the rest of this section.

### Figure 19.14. Connect Dialog

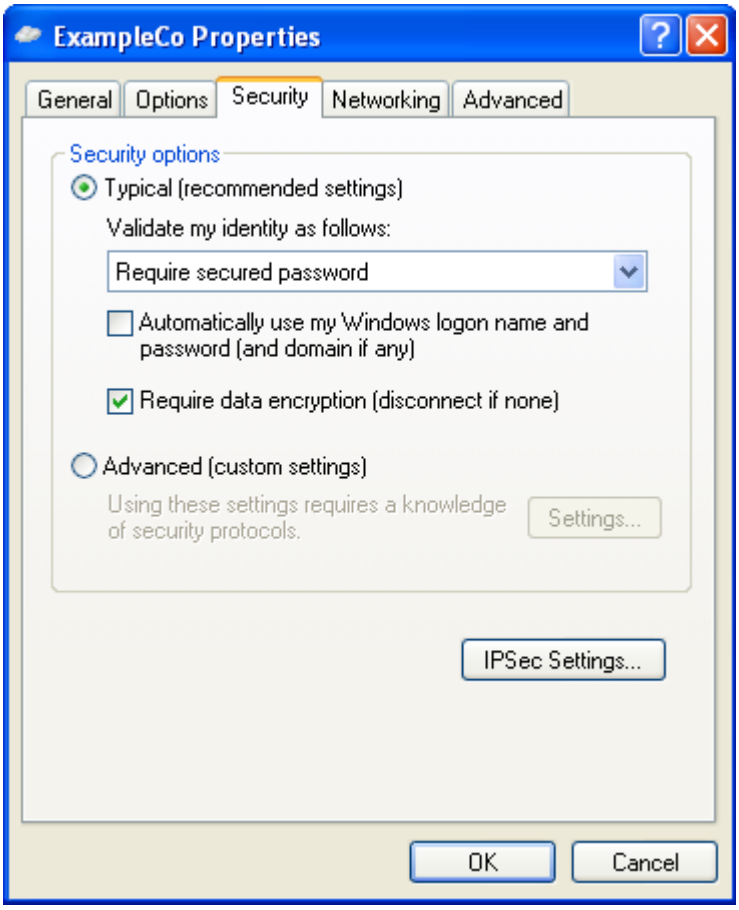

### Figure 19.15. Connection Properties




There are some other settings that needs checked and perhaps adjusted. From within Network Connections, right click on the icon for the PPTP connection, then click Properties (Figure 19.15, “Connection Properties”).

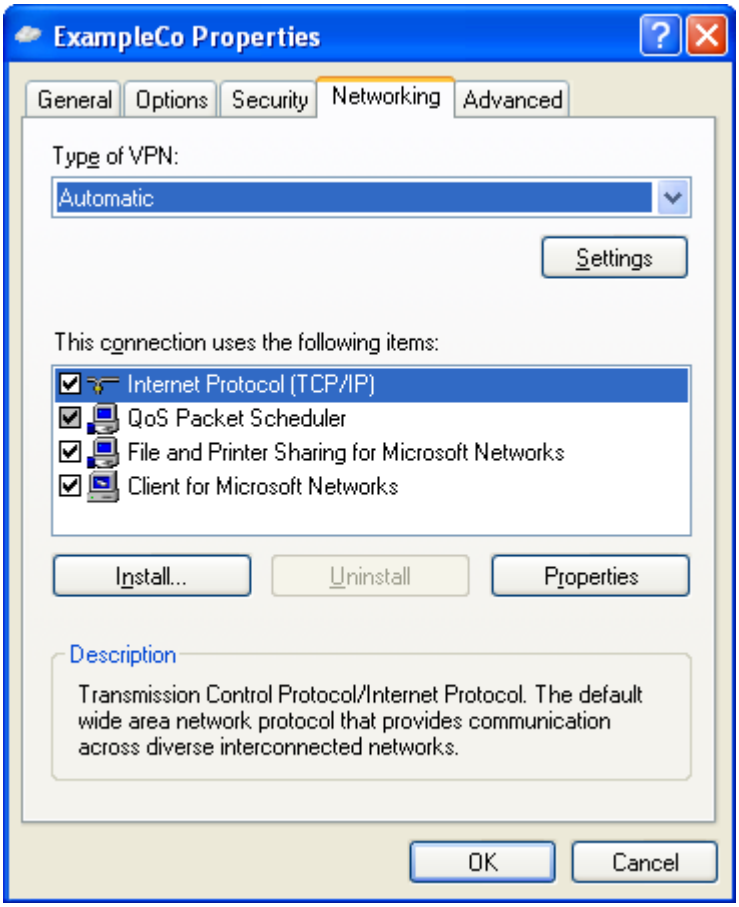
Click on the Security tab (Figure 19.16, “Security Tab”). Under Verify my identity as follows, make sure that Require secured password is chosen. Also ensure that Require data encryption (disconnect if none) is checked.

**Figure 19.16. Security Tab**



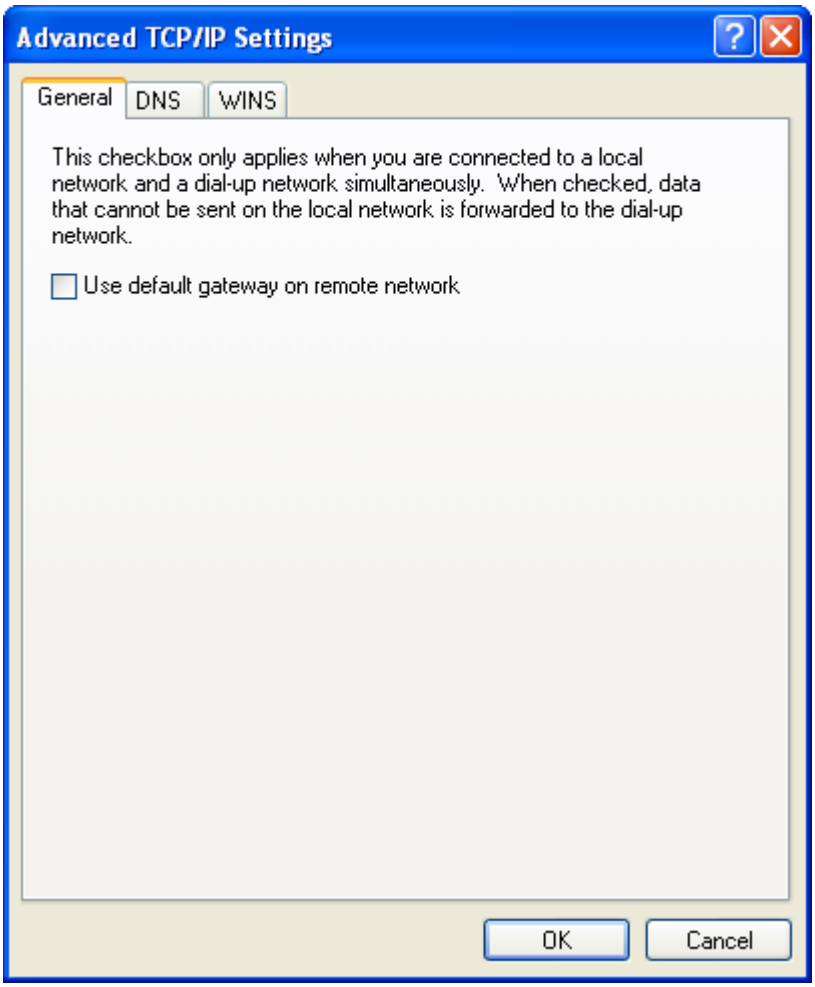

Now click on the Networking tab. As you can see in Figure 19.17, “Networking Tab”, the Type of VPN drop down defaults to **`Automatic`**. What this really means is "try stuff until something works." PPTP is the last thing Windows will try, and there will be a delay of up to 30 seconds or more while it waits for the other options to time out, so you likely want to select **`PPTP`** here to avoid that delay and any complications that may arise from Windows' automatic methodology.

**Figure 19.17. Networking Tab**




By default, this connection will send all traffic out through the PPTP connection as its gateway. This may or may not be desirable, depending on your intended configuration. This behavior is configurable, however. To change this, double click on Internet Protocol (TCP/IP), and click the Advanced button. Now uncheck Use default gateway on remote network as in Figure 19.18, “Remote Gateway Setting”, then click OK on all the open windows. With this option unchecked, only traffic bound for the subnet of the PPTP connection will traverse the tunnel.

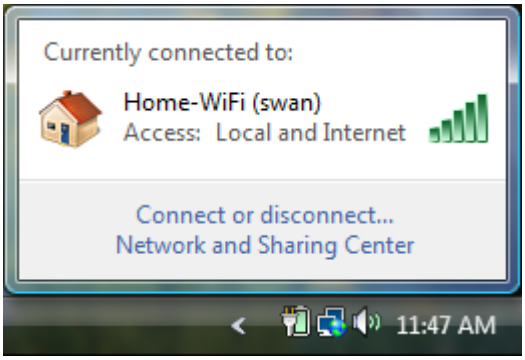
**Figure 19.18. Remote Gateway Setting**




Now the PPTP connection will only send traffic destined for its subnet across the VPN. If you need to selectively route traffic, see the section called “PPTP Routing Tricks”.

# Windows Vista

**Figure 19.19. Vista Network Connections**

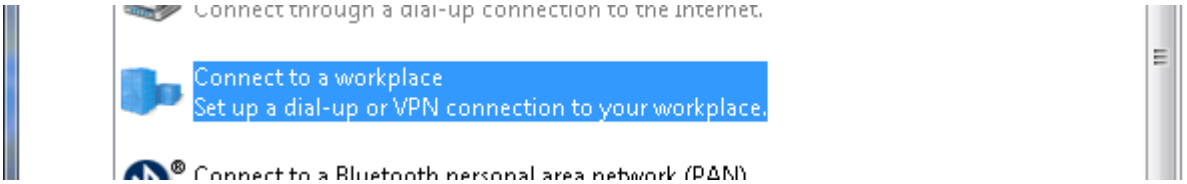



Click on the Network Connection indicator icon in the system tray by the clock, then click Connect or Disconnect as seen in Figure 19.19, “Vista Network Connections”.

Click Set up a connection or network (Figure 19.20, “Setup A Connection”), then click Connect to a workplace (Figure 19.21, “Connect to a Workplace”) and then Next.

### Figure 19.20. Setup A Connection

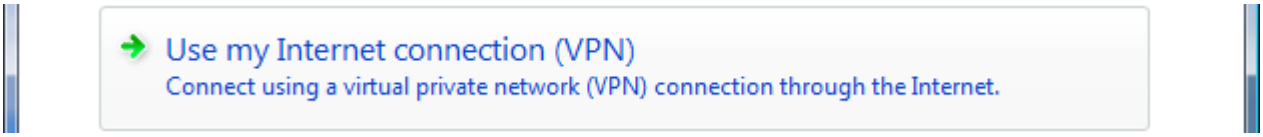

### Figure 19.21. Connect to a Workplace




If prompted, choose No, create a new connection, and click Next.

Click Use my Internet connection (VPN) (Figure 19.22, “Connect using VPN”).

### Figure 19.22. Connect using VPN

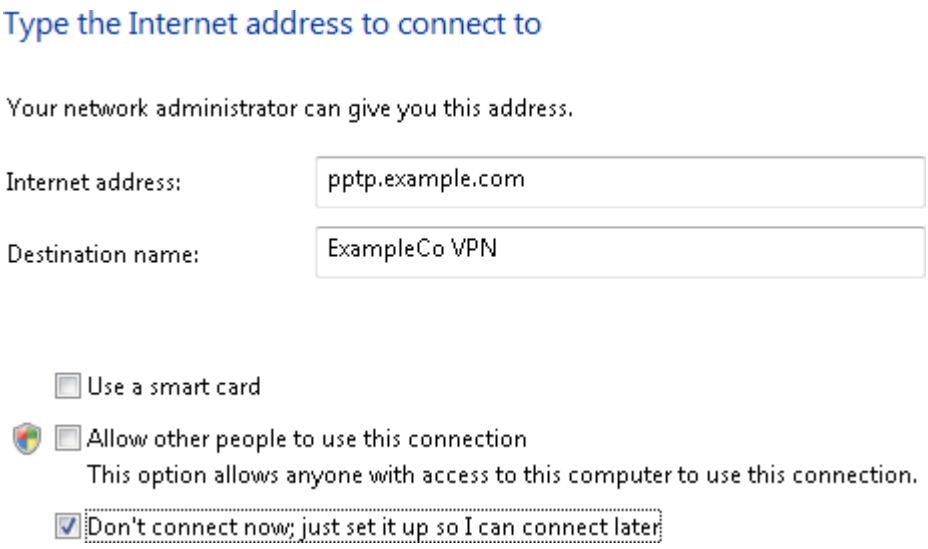



On the next screen, shown in Figure 19.23, “Connection Setup”, enter the WAN IP of the remote pfSense router under Internet Address.

Enter a name for the connection under Destination name.

Check Don't Connect Now and click Next.

### Figure 19.23. Connection Setup




Enter the username and password, as in Figure 19.24, “Authentication Settings”, then click Create. A screen like Figure 19.25, “Connection is Ready” should appear indicating that the connection has been created.

**Figure 19.24. Authentication Settings**

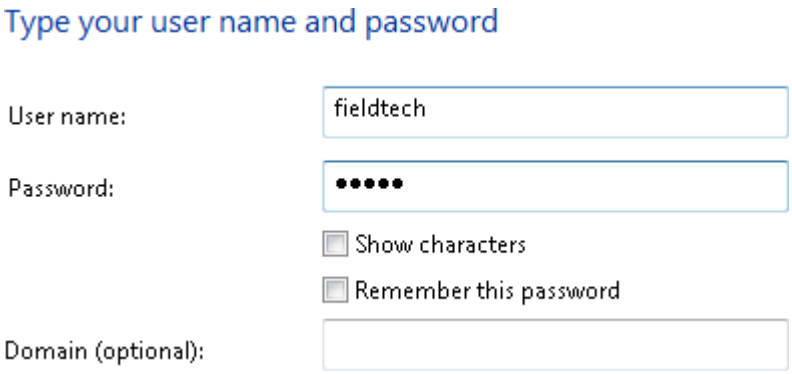
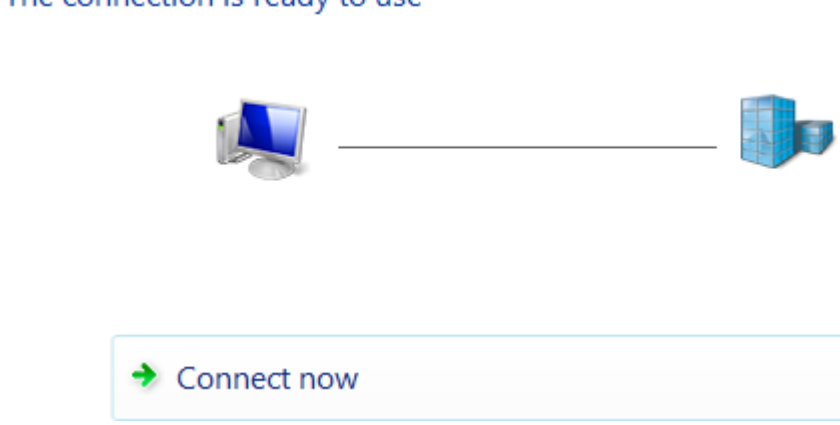


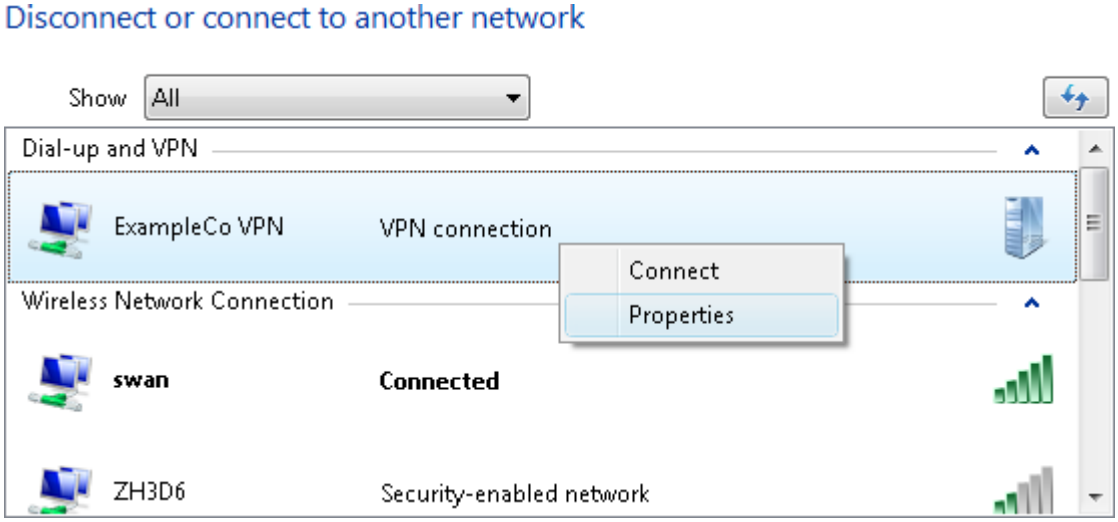

**Figure 19.25. Connection is Ready**




You should now have a PPTP dial-up entry that works like any other dial-up connection. Quickly access this by clicking the Network Connection indicator icon in the system tray, click Connect or Disconnect, choose the VPN connection, and click Connect.

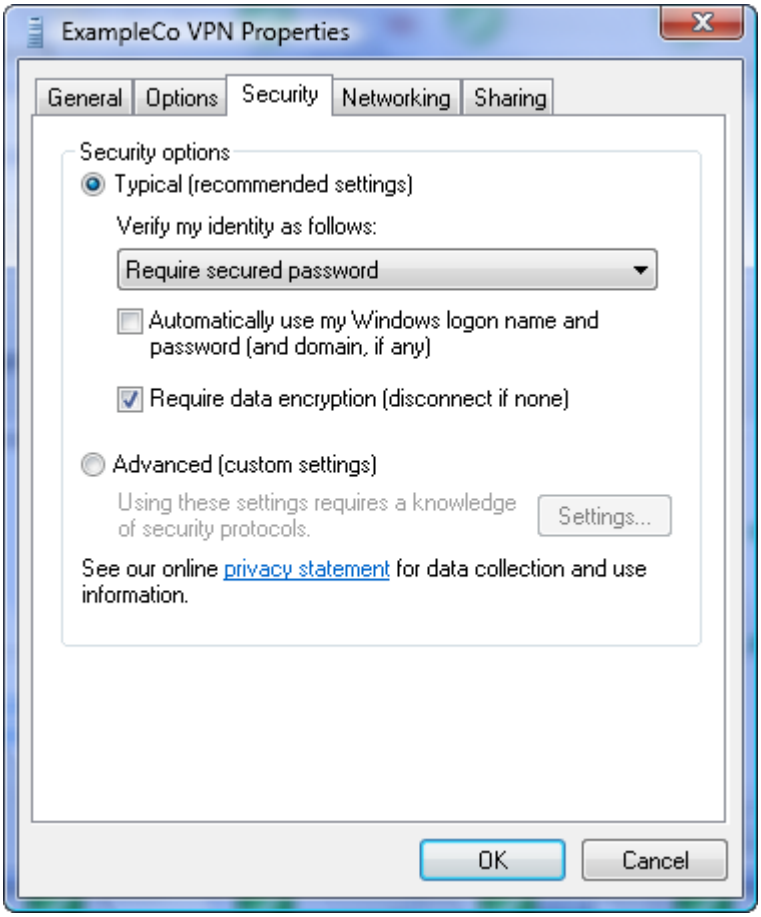
However, before connecting for the first time, there are some other settings to double check. First, click the Network Connection indicator icon in the system tray, and click Connect or Disconnect. Right click on the VPN connection that was just created, then click Properties as demonstrated in Figure 19.26, “Get Connection Properties”.

**Figure 19.26. Get Connection Properties**



Change to the Security tab (Figure 19.27, “VPN Security Settings”). Under Verify my identity as follows, make sure that Require secured password is chosen. Also ensure that Require data encryption (disconnect if none) is checked.

### Figure 19.27. VPN Security Settings




Now change to the Networking tab (Figure 19.28, “VPN Networking Settings”). It is probably best to uncheck Internet Protocol Version 6 (TCP/IPv6) at this point.

The Type of VPN drop down defaults to **`Automatic`**. What this really means is "try stuff until something works." PPTP is the last thing Windows will try, and there will be a delay of up to 30 seconds or more while it waits for the other options to time out, so you likely want to select **`PPTP`** here to avoid that delay and any complications that may arise from Windows' automatic methodology.

As with Windows XP, this connection will send all traffic out through the PPTP connection as its gateway. This may or may not be desirable, depending on your intended configuration. If you want all traffic to go across the tunnel, skip the rest of this section. Otherwise, click on Internet Protocol Version 4 (TCP/IPv4), then click Properties.

**Figure 19.28. VPN Networking Settings**

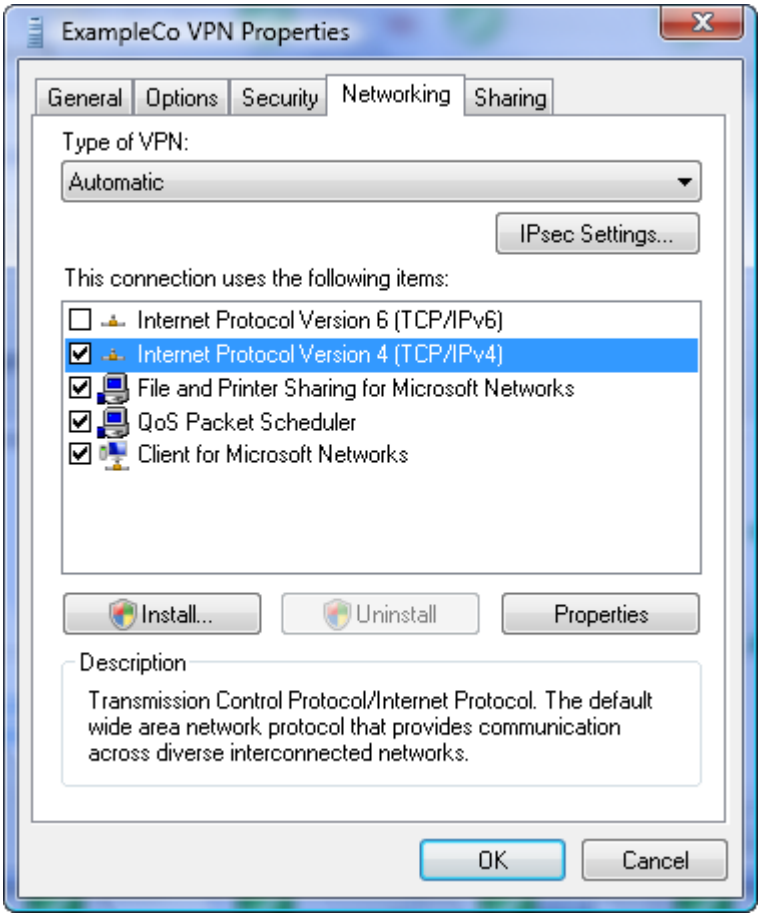
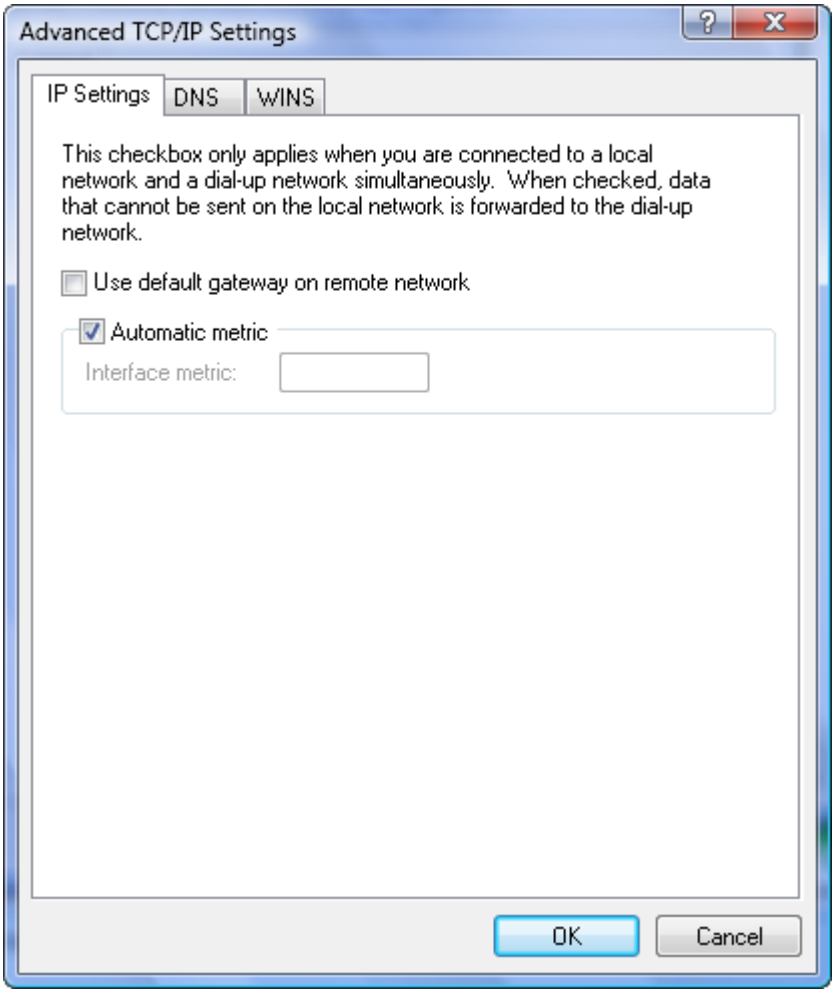


**Figure 19.29. VPN Gateway**




Click the Advanced button, and then uncheck Use default gateway on remote network as shown in Figure 19.29, “VPN Gateway”. Click OK or Close on all the windows that were just opened.

Now the PPTP connection will only send traffic destined for its subnet across the VPN. If you need to selectively route traffic, see the section called “PPTP Routing Tricks”.

# Windows 7

The PPTP client setup procedure in the release version (RTM) of Windows 7 is virtually identical to Windows Vista.

# Mac OS X

Open System Preferences, then click View → Network. Click the plus at the bottom of the list of your network adapters to add a new connection, which can be seen in Figure 19.30, “Add network connection”.

**Figure 19.30. Add network connection**


In the Interface drop down, select VPN, and for VPN Type select PPTP. Fill in the service name as desired and click Create. These choices are shown in Figure 19.31, “Add PPTP VPN connection”

**Figure 19.31. Add PPTP VPN connection**

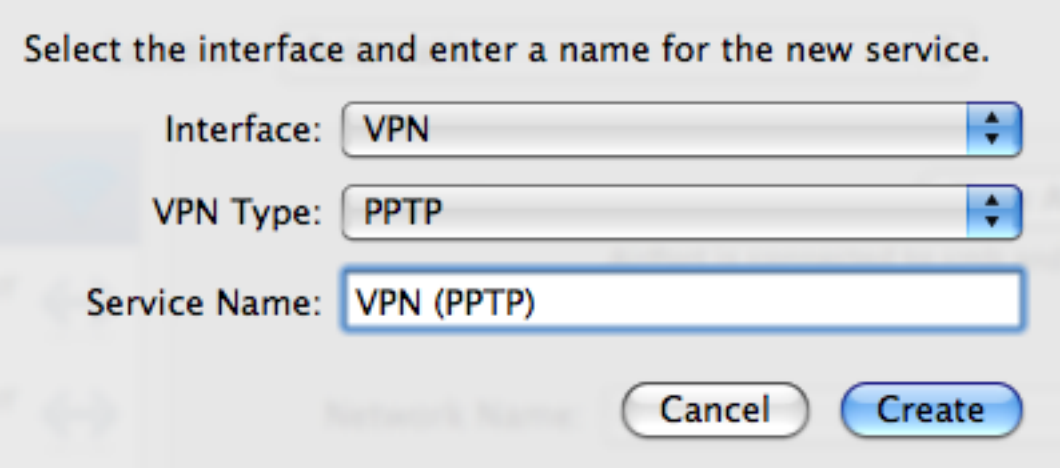
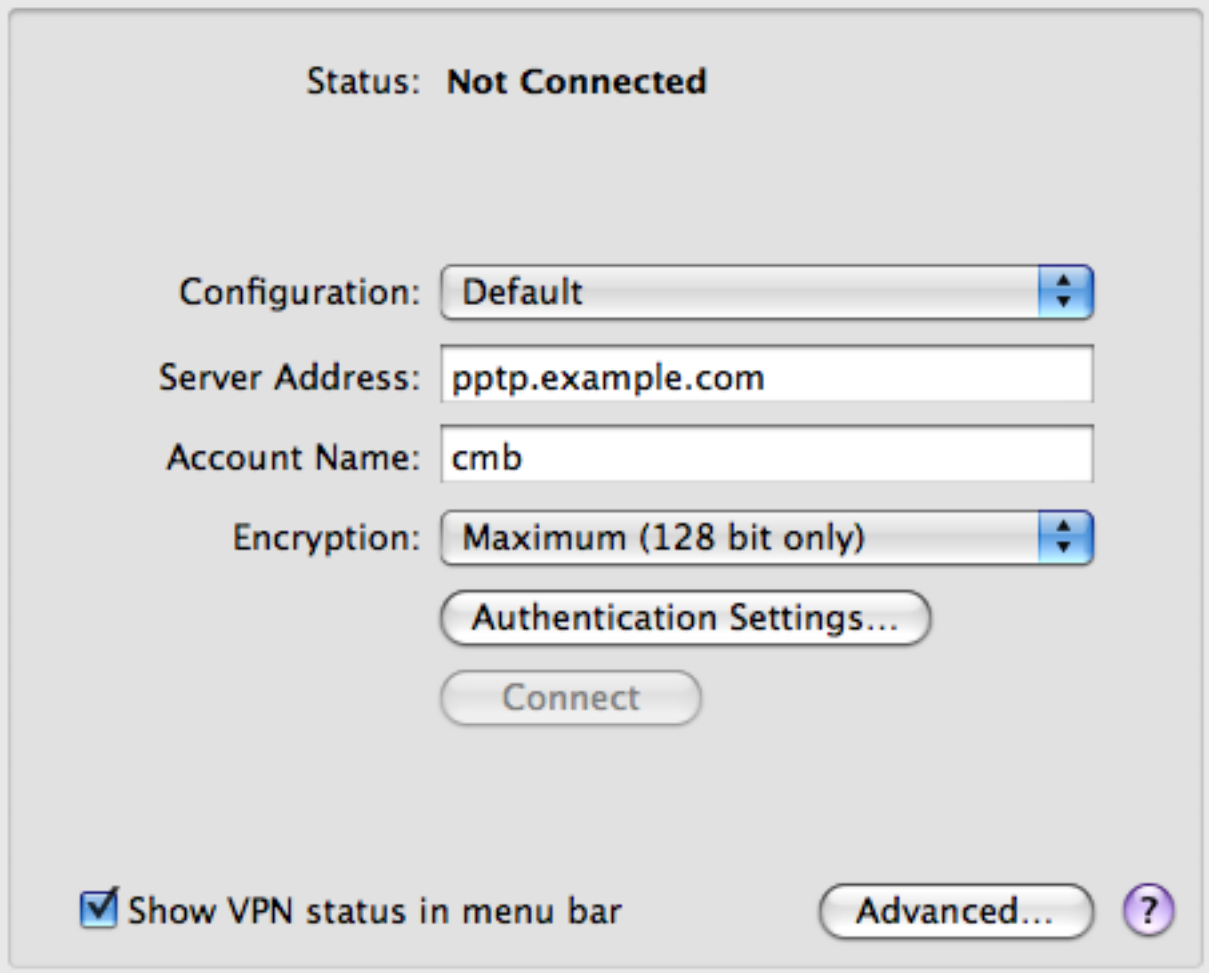


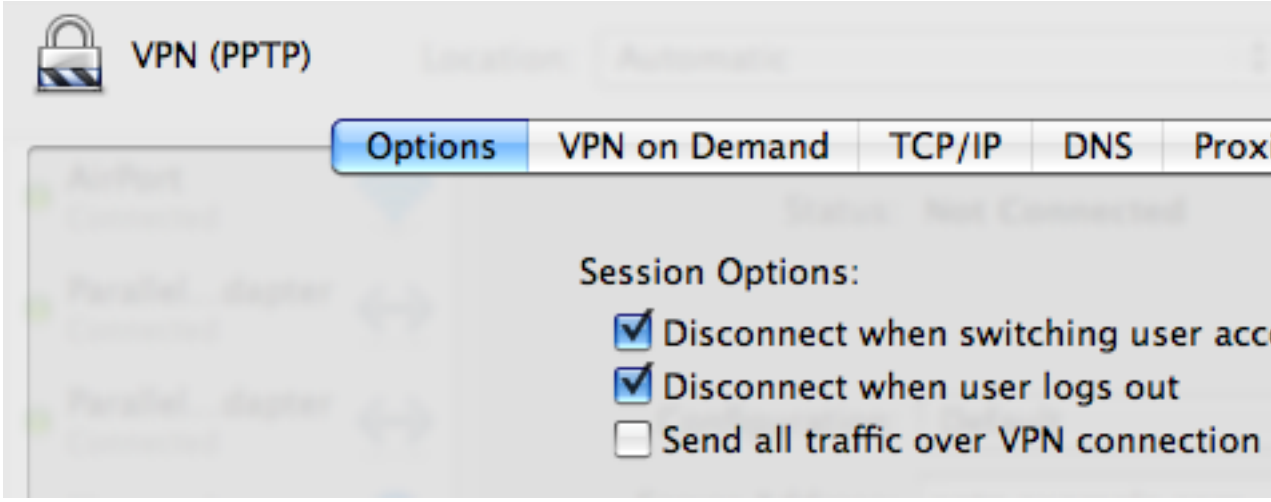
**Figure 19.32. Configure PPTP VPN connection**




This will take you back to the Network screen where you finish configuration for the PPTP VPN connection. Fill in the server address, account name, and choose Maximum (128 bit only) for Encryption. An example is shown in Figure 19.32, “Configure PPTP VPN connection”. Then click the Advanced button.

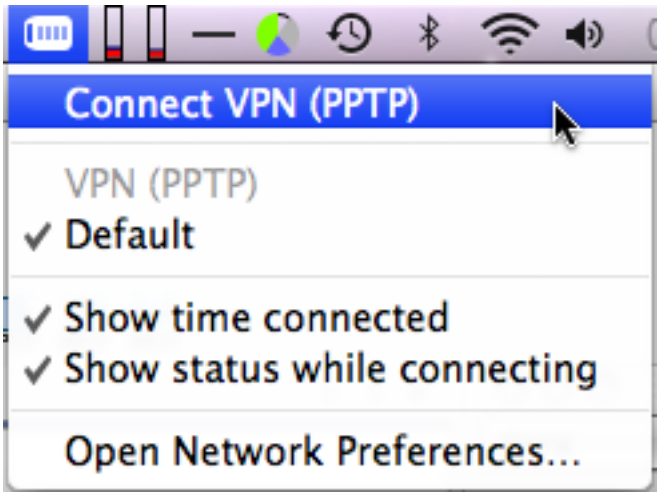
The Advanced screen has a number of options, some of them shown in Figure 19.33, “Advanced options”, though only one you may want to consider changing. The Send all traffic over VPN connection box is unchecked by default. If you want all the traffic from the client to traverse the VPN while connected, check this box. Click OK when finished.

**Figure 19.33. Advanced options**




Since I checked Show VPN status in menu bar as shown in Figure 19.32, “Configure PPTP VPN connection”, my connection now shows at the top of the screen. To connect, click on the name of your connection like that seen in Figure 19.34, “Connect to PPTP VPN”.

**Figure 19.34. Connect to PPTP VPN**




# PPTP Redirection

PPTP redirection allows you to forward PPTP traffic destined to your WAN IP to an internal PPTP server. To enable it, select Redirect incoming PPTP connections to and enter your internal PPTP server's IP in the PPTP redirection box. This is functionally equivalent to adding Port Forward entries for TCP port 1723 and the GRE protocol to your internal PPTP server, which you can do instead if you prefer. Its existence is largely a hold over from m0n0wall, where the underlying IPFilter does not support forwarding the GRE protocol. It has been retained because of m0n0wall users' familiarity with the feature, and some users prefer the ease of a single entry rather than two port forward entries.

Firewall rules for the GRE protocol and TCP port 1723 are automatically added on the WAN. You do not need to enter any firewall rules when using PPTP redirection, unless you have Disable all auto-added VPN rules checked under System → Advanced.

# PPTP Troubleshooting

This section covers troubleshooting steps for the most common problems users encounter with PPTP.

# Cannot connect

First, ensure the client computer is connected to the Internet. If that succeeds, make note of the error you are receiving from the client. Windows (except Vista) will provide an error code that will help considerably in narrowing down the potential problems. Windows Vista removed this and hence makes it difficult to properly troubleshoot connection failures, but thankfully the same error codes that have been around for over a decade are back in Windows 7. Troubleshooting with Vista is not recommended.

For those using non-Windows clients, the troublesome areas are generally the same, though you may have to try them all to determine the specific problem.

## Error 619

Error 619 means something along the way is breaking your GRE traffic. This is almost always caused by the firewall in front of the client. If the client is also behind pfSense, first ensure that none of the scenarios outlined in the section called “PPTP Limitations” apply. If the firewall the client is behind is another product, you may need to enable PPTP passthrough or a similar setting for PPTP to function, if it can at all. In some cases, like 3G wireless providers assigning private IPs to customers, you will be stuck with choosing another VPN option.

## Error 691

Error 691 is caused by an invalid username or password. This means the user is not entering the correct username or password in the PPTP client. Correct the username and/or password, matching up with the information configured in the local user database for PPTP, or on the RADIUS server.

## Error 649

You may see error 649 when authenticating to RADIUS on a Microsoft Windows server using IAS/NPS. This means the account does not have permission to dial in, and the cause will likely be one of three things.

1. Dial in permission set to "Deny access" — go to the properties of the user's account in Active Directory Users and Computers and click the Dial-in tab. Depending on your preferred IAS or NPS configuration, you will want either **`Allow access`** or **`Control access through remote access policy`**.

2. User's password is expired — if the user's password is expired, they cannot log in over PPTP.

3. Incorrect IAS/NPS configuration — you may have configured remote access policies in IAS or NPS as such that users are not authorized to be connected.

# Connected to PPTP but cannot pass traffic

Ensure you have added firewall rules to the PPTP VPN interface as described in the section called “Configure firewall rules for PPTP clients”.

Also ensure the remote subnet across the VPN is different from the local subnet. If you are trying to connect to a 192.168.1.0/24 network across VPN and the local subnet where the client is connected 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's important to choose a relatively obscure LAN subnet when using VPN, as discussed in the section called “LAN Interface Configuration”.

# PPTP Routing Tricks

If you only want selected subnets to be routed across the PPTP tunnel, it can still be done with some custom route commands on the client. The following technique works under Windows XP, Vista, and

Windows 7, but can probably be altered to work on most any platform. This assumes that you have already configured the client to not send all traffic across the connection (i.e. not using the remote gateway).

First, the PPTP client needs to be assigned a static address in the user profile. This can be done using the built-in authentication, or via RADIUS. This static address should be outside of the general assignment pool since this is not a reservation.

The trick is to route traffic destined for the remote subnets to the assigned PPTP address. This will cause traffic for those subnets to ride the tunnel to the other side. It isn't limited to subnets that are immediately reachable on the other side, either, as any subnet can be used. This is handy if you want to also route access to a third-party site through the VPN tunnel as well.

These commands can be typed at a command line, but are more at home in a batch file as in this example:

```
@echo off
route add 192.168.210.0 mask 255.255.255.0 192.168.1.126
route add    10.99.99.0 mask 255.255.255.0 192.168.1.126
route add    172.16.1.0 mask 255.255.252.0 192.168.1.126
pause
```

In that example, *192.168.1.126* is the static IP assigned to this particular PPTP client's username . These commands would route the three specified subnets across the PPTP connection, in addition to the subnet for the connection itself. The pause is optional, but may help to ensure that all the routes were added successfully. The batch file will need to be run each time the connection is established.

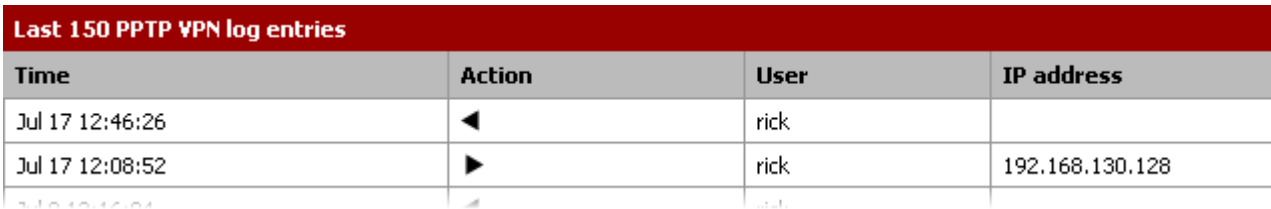
### Note

On Windows Vista and Windows 7, these commands will need to be run as Administrator. If you created a shortcut to this batch file, its properties may be altered so it always runs in that manner. Alternately, you could right click on the batch file and choose Run As Administrator.

# PPTP Logs

A record of login and logout events is kept on Status → System Logs, on the VPN tab under PPTP Logins.

**Figure 19.35. PPTP Logs**



Last 150 PPTP VPN log entries

| Time | Action | User | IP address |
|---|---|---|---|
| Jul 17 12:46:26 | ◀ | rick | |
| Jul 17 12:08:52 | ▶ | rick | 192.168.130.128 |

As seen in Figure 19.35, “PPTP Logs”, each login and logout should be recorded with a timestamp and username, and each login will also show the IP address assigned to the PPTP client. The full log can be found on the PPTP Raw tab.

# Chapter 20. L2TP VPN

pfSense can act as a L2TP VPN server as one of its VPN options. L2TP is purely a tunneling protocol that offers no encryption of its own, so it's typically combined with some other encryption technique, such as IPsec. Currently in pfSense it's not possible to use L2TP+IPsec together in the way most clients, like those built into Windows, OS X and iOS, expect.

For general discussion of the various types of VPNs available in pfSense and their pros and cons, see Chapter 16, *Virtual Private Networks*.

# L2TP Security Warning

L2TP is not encrypted, so it should never be used on its own for traffic that should be private. Some devices, such as Android, offer an L2TP-only client which is capable of connecting back to pfSense but it should only be used if you are either transmitting traffic that is already encrypted, or if you are transmitting traffic you do not consider private. For example, you might use it to tunnel Internet traffic so it appears to originate from another location but flow unencrypted.

# L2TP and Firewall Rules

By default, when you have the L2TP server enabled, firewall rules will *NOT* be automatically added to the chosen interface to permit TCP port 1701. You will need to add a firewall rule to whichever interface the L2TP traffic will be entering.

# L2TP and Multi-WAN

Because L2TP only relies on a single TCP port (1701), the server can run on any interface you choose, so it is fully multi-WAN compatible..

# L2TP Limitations

Most people would expect to use L2TP in combination with IPsec, but due to the way that L2TP+IPsec clients work, that would require altering our IPsec daemon in a way that would compromise its overall security. We're working to find a way around that for a future release, but for now it's not yet possible.

# L2TP Server Configuration

If you still wish to use L2TP, first browse to VPN → L2TP. Select Enable L2TP server.

## Interface

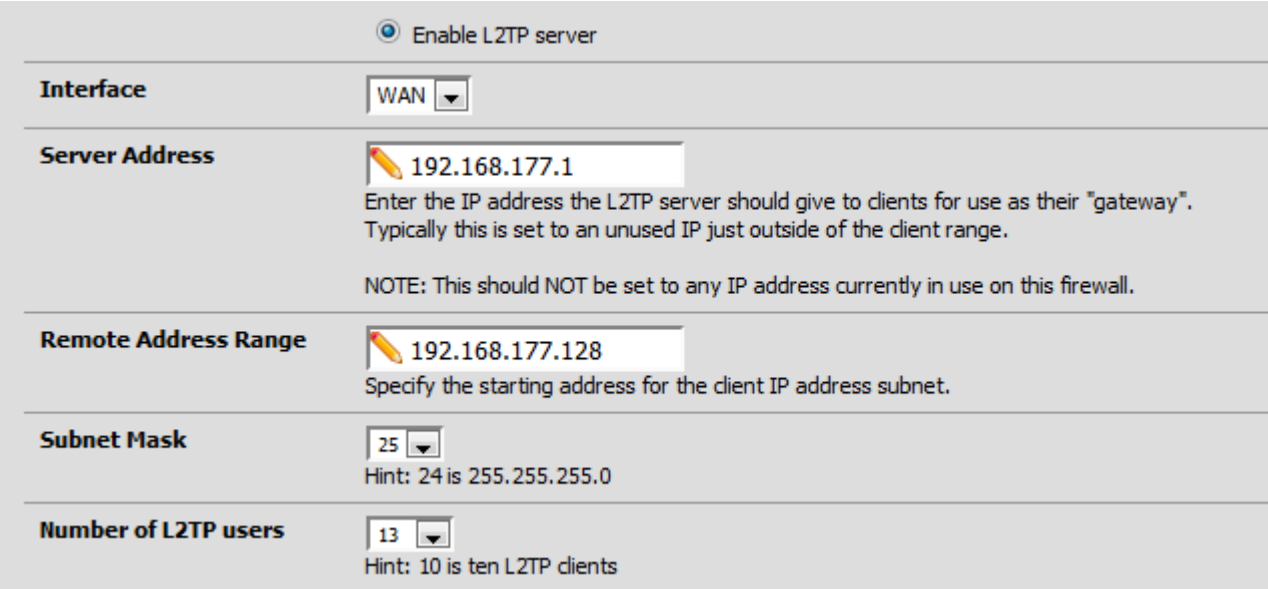
The Interface setting controls where the L2TP daemon will bind and listen for connections. Normally this would be whatever WAN interface would be accepting the inbound connections.

## IP Addressing

You will need to decide what IP addresses to use for the L2TP server and clients, and now many concurrent clients you want to support. The Number of L2TP users field controls how many L2TP users will be allowed to connect at the same time, in this example we selected **16**. The Remote Address Range is usually a new and unused subnet, such as 192.168.177.128/25 (.128 through .255). These are

the addresses to be assigned to clients when they connect. Then select an IP address outside of that range for the Server Address, such as 192.168.177.1 as shown in Figure 20.1, “L2TP IP Addressing”.

**Figure 20.1. L2TP IP Addressing**




# Authentication

In some L2TP implementations, a Secret is required as sort of a group password or pre-shared key. Support for this varies from client to client, so if you're in doubt, leave it blank.

The Authentication Type setting lets you choose between **PAP** or **CHAP** authentication for users. Support for this can vary from client to client and it may also depend on your RADIUS server as well. **CHAP** is more secure, but **PAP** is more widely compatible.

You can authenticate users from the local user database, or via RADIUS. RADIUS allows you to connect to another server on your network to provide authentication. This can be used to authenticate L2TP users from Microsoft Active Directory (see the section called “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. You can also add a second RADIUS server to use in case the first one fails. For authentication using the local user database, leave that box unchecked. You will have to add your users on the Users tab of the VPN → L2TP screen unless using RADIUS. See the section called “Adding Users” below for more details on the built-in authentication system.

# Save changes to start L2TP server

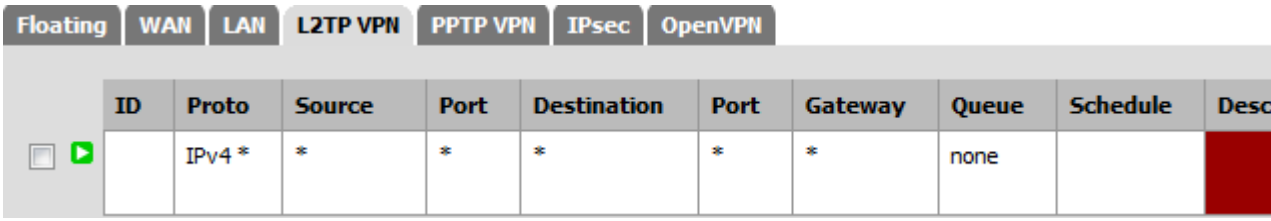
After filling in the aforementioned items, click Save. This will save your configuration and launch the L2TP server. If you are authenticating your users with the local user database, click the Users tab and enter your users there.

# Configure firewall rules for L2TP clients

Browse to Firewall → Rules and click the L2TP VPN tab. These rules control what traffic is permitted from L2TP clients. Until you add a firewall rule here, all traffic initiated from connected L2TP clients will be blocked. Traffic initiated from your LAN to the L2TP clients is controlled using your LAN firewall rules. Initially you may want to add an allow all rule here for testing purposes as shown in

Figure 20.2, “L2TP VPN Firewall Rule”, and once you verify functionality, restrict the ruleset as desired.

### Figure 20.2. L2TP VPN Firewall Rule




# Adding Users

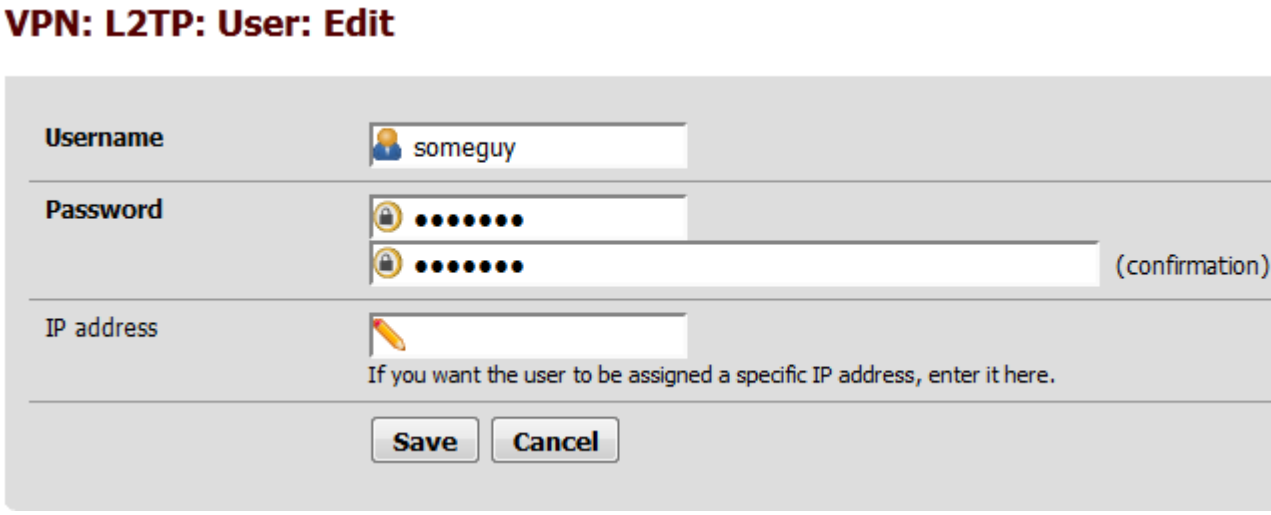
Adding users via RADIUS will vary from one implementation to another. This fact makes it beyond the scope of this section, but should be covered in the documentation for the particular RADIUS server being employed.

Adding users to pfSense's built-in L2TP users system is quite easy. First, click on VPN → L2TP, and then the Users tab. You will be presented with an empty users screen as shown in Figure 20.3, “L2TP Users Tab”. Click the button to add a user.

### Figure 20.3. L2TP Users Tab

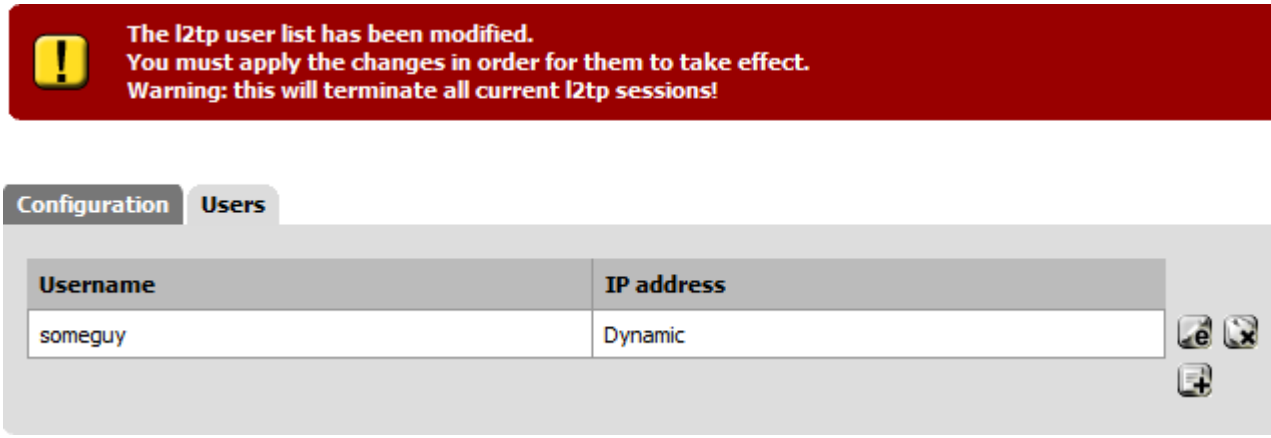

After clicking , the user editing page will appear. Fill it in with the username and password for a user, as in Figure 20.4, “Adding a L2TP User”. You may also enter a static IP assignment if desired.

### Figure 20.4. Adding a L2TP User




Click Save, and then the user list will return (Figure 20.5, “Applying L2TP Changes”), but before the change will take effect, the Apply Changes button must first be clicked.

**Figure 20.5. Applying L2TP Changes**




Repeat that process for each user you would like to add.

If you need to edit an existing user, click . Users may be deleted by clicking .

# L2TP Troubleshooting

This section covers troubleshooting steps for the most common problems users encounter with L2TP.

## Cannot connect

Check that you have added firewall rules to the external interface where the L2TP traffic will enter the firewall. Also make sure you're connecting to the interface IP address chosen on the L2TP settings.

## Connected to L2TP but cannot pass traffic

Ensure you have added firewall rules to the L2TP VPN interface as described in the section called “Configure firewall rules for L2TP clients”.

Also ensure the remote subnet across the VPN is different from the local subnet. If you are trying to connect to a 192.168.1.0/24 network across VPN and the local subnet where the client is connected 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's important to choose a relatively obscure LAN subnet when using VPN, as discussed in the section called “LAN Interface Configuration”.

# 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 should be 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.

# Chapter 21. Traffic Shaper

Traffic shaping, or network Quality of Service (QoS), is a means of prioritizing the network traffic traversing your firewall. Without traffic shaping, packets are processed on a first in/first out basis by your 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. The traffic shaper wizard in pfSense gives you the ability to quickly configure QoS for common scenarios, and custom rules may also be created for more complex tasks. For simplicity, the traffic shaping system in pfSense may also be referred to as the "shaper", and the act of traffic shaping may be called "shaping".

pfSense 2.x introduced a separate shaper concept called Limiters. Limiters allow you to set hard bandwidth limits for either a group or on a per-IP basis, and inside of those bandwidth limits you may also optionally assign priorities to traffic.

# Traffic Shaping Basics

For those of you who are unfamiliar with traffic shaping, it is sort of 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, some regular packets may need to be tossed out to keep the place from getting too crowded.

The way that shaping is accomplished in pf, and thus pfSense, may be a little counter-intuitive at first because the traffic has to be limited in a place where pfSense can actually control the flow. Incoming traffic from the Internet going to a host on the LAN (downloading) is actually shaped coming *out* of the LAN interface from the pfSense system. In the same manner, traffic going from the LAN to the Internet (uploading) is shaped when *leaving* the WAN.

There are traffic shaping queues, and traffic shaping rules. The queues are where bandwidth and priorities are actually allocated. 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 2.x, shaper rules are mostly handled on the Floating tab using the **`match`** action that only assigns the traffic into queues, but you can also assign traffic into queues using pass rules on any other interface as well.

Limiter rules are handled differently. Limiters apply on regular pass rules and enforce their limits on the traffic as it enters and leaves the interface. When working with limiters, you almost always work them in pairs, one for the "download" direction traffic and one for the "upload" direction traffic.

# What the Traffic Shaper can do for you

The basic idea of traffic shaping, raising and lowering the priorities of packets, or simply keeping them under a certain speed, is a simple one. 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 our users.

## Keep Browsing Smooth

Asymmetric links, where the download speed differs from the upload speed, are commonplace these days, especially with DSL. Some links are so out of balance that the maximum download speed is almost unattainable because it is difficult 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, TCP's congestion control mechanisms will kick in and slow down the connection.

You may have noticed this situation before: When uploading a file over such a link, browsing and downloading slows to a crawl or stalls. This happens because the uploading portion of the circuit is full

from the file upload, there is little room to send ACK packets which allow downloads keep flowing. By using the shaper to prioritize ACK packets, you can achieve 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 desriable if the available outgoing bandwidth is heavily utilized.

## Keep VoIP Calls Clear

If your Voice over IP calls use the same circuit as data, then uploads and downloads may degrade your call quality. pfSense can prioritize the call traffic above other protocols, and ensure that the calls make it through clearly without breaking up, even if you're streaming hi-def video from Netflix at the same time. Instead of the call breaking up, the speed of the other transfers will be reduced to leave room for the calls.

## Reduce Gaming Lag

There are also options to give priority to the traffic associated with network gaming. Similar to prioritizing VoIP calls, the effect is that even if you are downloading while playing, the response time of the game should still be nearly as fast as if the rest of your connection were idle.

## Keep P2P Applications In Check

By lowering the priority of traffic associated with known peer-to-peer ports, you can rest easier knowing that even if those programs are in use, they won't hinder other traffic on your 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

Using limiters you can apply a bandwidth limit to a group of people, such as all traffic on an interface, or you can set masking on the limiters to apply them on a per-IP basis. This way you 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 shaping. The following network cards are capable of using traffic shaping, according to the man page for `altq(4)`:

`age(4)`, `alc(4)`, `ale(4)`, `an(4)`, `ath(4)`, `aue(4)`, `awi(4)`, `bce(4)`, `bfe(4)`, `bge(4)`, `bridge(4)`, `bwn(4)`, `cas(4)`, `dc(4)`, `de(4)`, `ed(4)`, `em(4)`, `ep(4)`, `fxp(4)`, `gem(4)`, `hme(4)`, `igb(4)`, `ipw(4)`, `iwi(4)`, `jme(4)`, `l2tp`, `lem(4)`, `le(4)`, `msk(4)`, `mxge(4)`, `my(4)`, `ndis(4)`, `nfe(4)`, `ng(4)`, `npe(4)`, `nve(4)`, `ovpnc`, `ovpns`, `pppoe`, `ppp`, `pptp`, `ral(4)`, `re(4)`, `rl(4)`, `rum(4)`, `run(4)`, `sf(4)`, `sis(4)`, `sk(4)`, `ste(4)`, `stge(4)`, `tun(4)`, `txp(4)`, `udav(4)`, `ural(4)`, `vge(4)`, `vlan(4)`, `vr(4)`, `vtnet(4)`, `wi(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 you have a card that does not support `ALTQ`, you may still use limiters instead.

# ALTQ Scheduler Types

In pfSense 1.2.x, there was only one ALTQ scheduler type, Hierarchical Fair Service Curve (HFSC). Now in pfSense 2.x there are more options available to cover a larger possible range of shaping

scenarios. The new options for ALTQ are Class-Based Queueing (CBQ), which can do bandwidth sharing between queues and bandwidth limits, and Priority Queueing (PRIQ), which only does prioritization.

Each of these is selectable in the shaper wizards, and the wizard will show the proper options and create the proper queues for you 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, then you may see a degradation of performance. Whether the loss is worse than working without shaping is up to your individual workload.

# Local LAN-to-LAN Traffic

In pfSense 1.2.x and before, the shaper could only apply to one LAN type interface, and the traffic between that LAN and other local networks would still be affected by the shaper and have its speed limited. In pfSense 2.x, the shaper now has a separate queue for this local traffic that will not limit it. Only traffic to and from the WAN to the LAN will be shaped.

# Hierarchical Fair Service Curve (HFSC)

The HFSC traffic shaping discipline has been in pfSense for many years. On 1.2.x, it was the only available choice. It is a very powerful discipline, good for ensuring that services like VoIP and video are delivered a minimum guaranteed amount of bandwidth.

In HFSC, the queues 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 the other disciplines. It is through these options that it achieves its guaranteed real-time processing and link sharing.

The Service Curve (sc) is where you can fine tune the bandwidth requirements for this queue.

- m1

  Burstable bandwidth limit

- d

  Time limit for bandwidth burst, specified in milliseconds. (e.g. 1000 = 1 second)

- m2

  Normal bandwidth limit

For instance, you need 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 timeframe d you 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 timeframe d, and m2 is the maximum that this discipline will allow to be used.

  **Note**

  The value for m2 cannot exceed 30% of the parent queue's available bandwidth.

- Link Share

  The bandwidth share of a backlogged queue. Will share the bandwidth between classes if the Real Time guarantees have been satisfied. If you set the m2 value for Link Share, it will override the Bandwidth setting for the queue. These two settings are the same, but if both are set, Link Share's m2 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 your traffic is governed as you see fit. For more information on m1, d, and m2 values for different scenarios, visit the pfSense Traffic Shaping forum [http://forum.pfsense.org/index.php/board,26.0.html].

# Class-Based Queueing (CBQ)

Class-Based Queueing, or CBQ, is similar to HFSC in that is can have a tree of queues nested under other queues. It supports bandwidth limits (not guarantees like HFSC), priorities for queues, and 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 you do not need to guarantee minimum bandwidths.

With CBQ, queue priorities range from 0 to 7, higher numbers indicating higher priority. Queues of an equal priority are processed in a round-robin fashion.

**Note**

Though the child queues can borrow from their parent queue, the sum of the bandwidth of the child queues cannot exceed the bandwidth of the parent, so this is not an alternative to using limiters to apply individual (e.g. per-IP) bandwidth limits.

## CBQ-Specific Queue Options

The CBQ discipline supports a the concept of borrow, meaning that if the checkbox on the queue is checked to Borrow from other queues when available, then it 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.

# Priority Queueing (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, on 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 pretty 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.

## CoDel Active Queue Management

The CoDel Active Queue Management (AQM) discipline was recently added to pfSense 2.1. The name is short for Controlled Delay and is pronounced "coddle". It was designed to combat some of the problems associated with bufferbloat in networking infrastructure. Bufferbloat is described in detail at http://www.bufferbloat.net/projects/bloat/wiki/Introduction. Basically, due to the size of buffers in network equipment, traffic can pile up and go in chunks rather than a smooth stream. 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 (512Kbps or less) and it does not gracefully handle large numbers of simultaneous flows or datacenter-grade traffic loads.

# Configuring the ALTQ Traffic Shaper With the Wizard

It is recommended that you configure the traffic shaper for the first time using the wizard, which will guide you through the process. Due to the complexity of the shaper queues and rules, it is not a good idea to attempt starting from scratch on your own. If you need custom rules, step through the wizard and approximate what you will need, then make the custom rules afterward. Each screen will setup unique queues, and rules that will control what traffic is assigned into those queues. Should you want to configure everything manually, simply specify your WAN speed at the first screen, then click Next through all the remaining screens without configuring anything.

### Note

Going through the wizard and clicking Finish at the end will replace all of your shaper queues and floating rules created by the wizard, or cloned from wizard rules, with the queues and rules from the new wizard configuration.

## Selecting a Wizard

To get started with the Traffic Shaping Wizard, click on Firewall → Traffic Shaper, and click the Wizards tab. You will be presented with a list of currently available shaper wizards. As of this writing, those included:

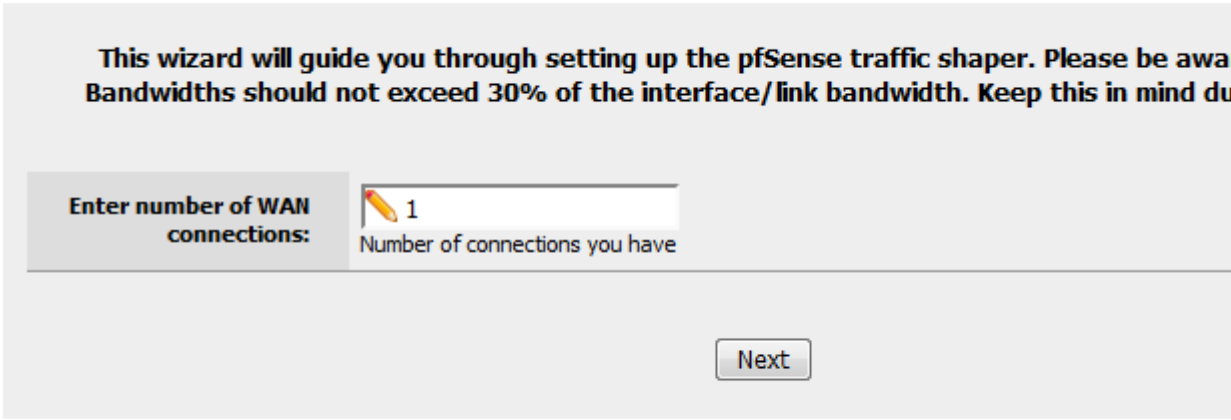
| | |
|---|---|
| Single LAN, Multi-WAN | Used when you have only one LAN, and one or more WANs. The "LAN" interface for shaping is assumed to be the interface indicated by "lan" in the pfSense interface assignments. The WAN interfaces are selectable. |
| Single WAN, Multi-LAN | Used when you have only one WAN, and one or more LANs. The "WAN" interface for shaping is assumed to be the interface indicated by "wan" in the pfSense interface assignments. The LAN interfaces are selectable. |
| Multiple LAN/WAN | Used when you have one or more WANs and one or more LANs. WAN and LAN interfaces are selectable. |
| Dedicated Links | Used when specific LAN+WAN pairings should be accounted for in the shaper configuration. WAN and LAN interfaces are selectable. |

Note that the first three are very similar, and you can, in fact, use any of them and end up with the same result. As such we will only cover one of those in this chapter. If you only have a single LAN and a single WAN, you can use any of the three and simply enter *1* when prompted for how many interfaces you have of the LAN/WAN variety.

# Starting the Wizard

Once you have selected the wizard you will use, and have clicked it, a the wizard starts and the first screen will then prompt for the number of connections that are variable in the wizard that was chosen. This step asks for the number of WANs, LANs, or both, as in Figure 21.1, “Entering the Interface Count”. Enter the number of connections that you have, as prompted, and proceed through the wizard. As you finish each screen of the wizard, click Next to continue to the following page.

**Figure 21.1. Entering the Interface Count**




# Networks and Speeds

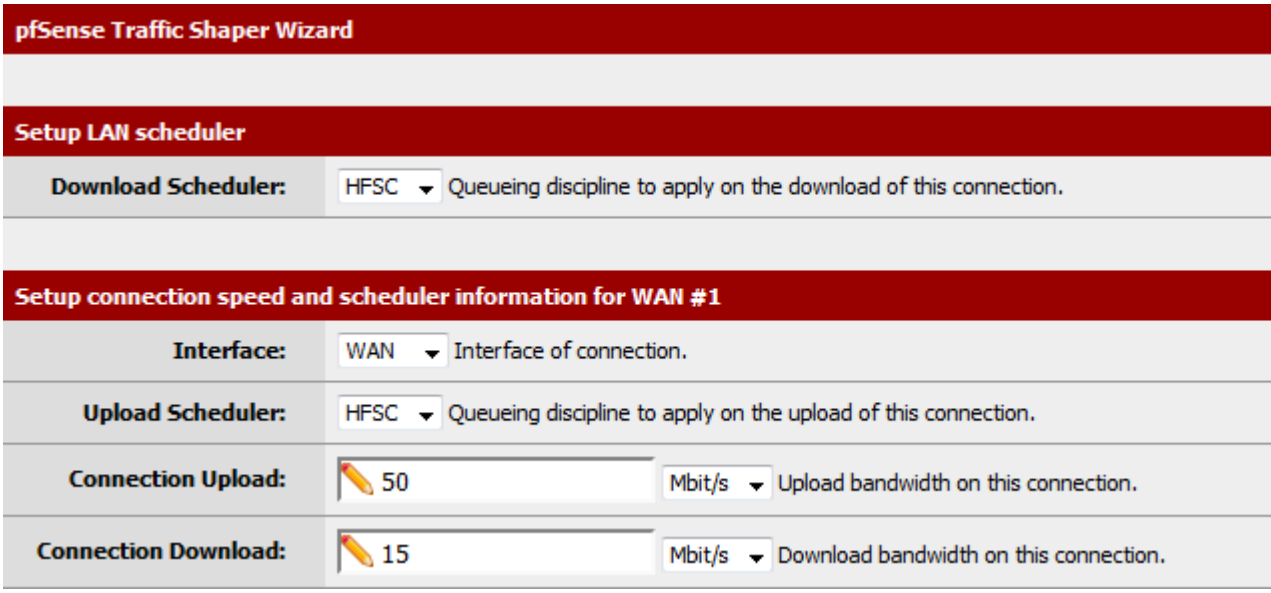
This screen, as shown in Figure 21.2, “Shaper Configuration”, is where you configure the network Interfaces that will be the Inside and Outside from the point of view of the shaper, along with the Download and Upload speeds for a given WAN. When there is more than one interface of a given type, there will be multiple sections on the page to handle each one individually.

Aside from the interfaces and their speeds, you also need to select an ALTQ Scheduler (the section called “ALTQ Scheduler Types”) for the WAN(s) and LAN(s). You will want to use the same scheduler on every interface.

Depending on your connection type, the true link speed may not be the actual usable speed. In the case of PPPoE, you have 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, you may lose as much as 13% of the advertised link speed. When in doubt of what to set the speed to, be a little conservative. Reduce by 10-13% and work your way back up to larger values. If you have a 3Mbit/s line, set it for about *2.7* **Mbit/s** and try it. You can always edit the resulting parent queue later and adjust the speed. If you set it low, the connection will be maxed out at exactly the speed you set. Keep nudging it up higher until you no longer get any performance gains.

The interface speeds can be specified in **Kbit/s**, **Mbit/s**, or **Gbit/s**.

**Figure 21.2. Shaper Configuration**




# Voice over IP

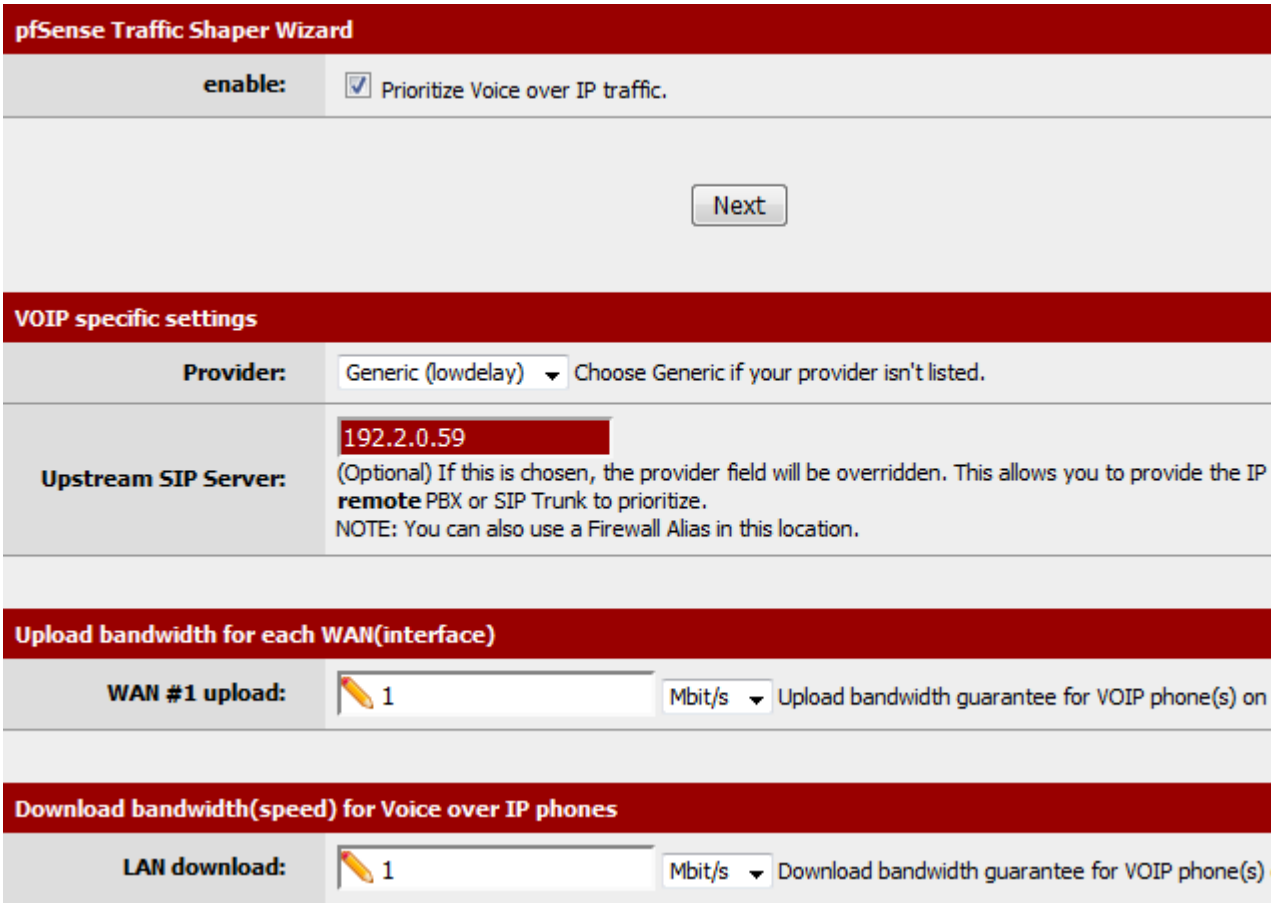
There are several options available for handling VoIP call traffic, shown in Figure 21.3, “Voice over IP”. The first choice, Prioritize Voice over IP traffic, is self-explanatory. It will enable the prioritization of VoIP traffic and this behavior can be fine-tuned by the other settings on the page. There are a few well-known providers including Vonage, Voicepulse, PanasonicTDA, and Asterisk servers. If you have a different provider, you can choose **`Generic`**, or override this setting with the Address field by entering the IP of your upstream PBX or SIP trunk, or an alias containing the IPs or networks for them.

You may also choose the amount of Bandwidth to guarantee for your VoIP phones. This will vary based on how many phones you have, and how much bandwidth each session will utilize.

## Note

The bandwidth reservation for a service such as VoIP cannot exceed 30% of the available bandwidth on the link. For example, on a 1Mbit/s link, you cannot reserve more than 300Kbit/s.

**Figure 21.3. Voice over IP**



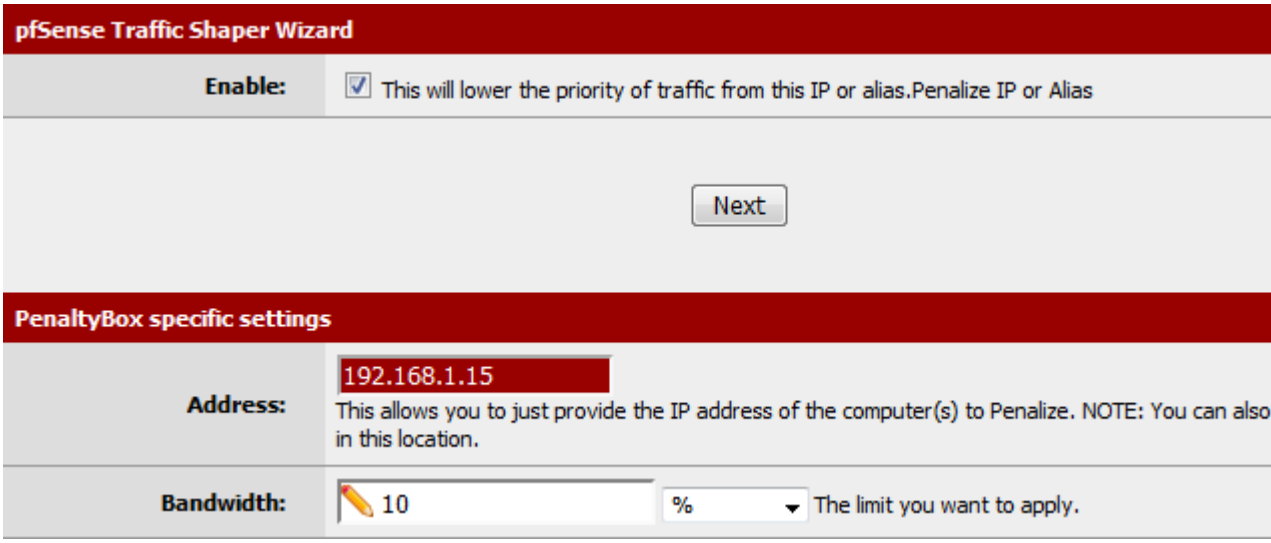

### Note

The way the shaper matches traffic with floating rules, it works best to use the remote SIP trunk or PBX because otherwise it may not be able to match the traffic properly. If you use the IPs of the phones it may only match traffic in one direction, or not at all.

# Penalty Box

The penalty box, depicted in Figure 21.4, “Penalty Box”, is a place to which you can relegate misbehaving users or devices that would otherwise consume more bandwidth than desired. These users are assigned a hard bandwidth cap which they cannot exceed. Check the Penalize IP or Alias to enable the feature, enter an IP or Alias in the Address box, then enter the Bandwidth limit, and choose the correct unit. Certain ALTQ schedulers may only allow you to select a percentage (**%**), others will let you set the value in **`Bit/s`**, **`Kbit/s`**, **`Mbit/s`**, or **`Gbit/s`**.

**Figure 21.4. Penalty Box**




# Peer-to-Peer Networking

The next screen, shown in Figure 21.5, “Peer-to-Peer Networking”, will let you set controls over many peer-to-peer (P2P) networking protocols. By design, P2P protocols will utilize all available bandwidth unless limits are put in place. If you expect P2P traffic on your network, it is a good practice to ensure that other traffic will not be degraded due to its use. To penalize P2P traffic, first check Lower priority of Peer-to-Peer traffic.

Many P2P technologies deliberately try to avoid detection. Bittorrent is especially guilty of this behavior. It often utilizes non-standard or random ports, or ports associated with other protocols. You can check the p2pCatchAll option which will cause any unrecognized traffic to be assumed as P2P traffic and its priority lowered accordingly. You can set hard bandwidth limits for this traffic underneath the catchall rule. The upload and download bandwidth limits are set in Kilobits per second.

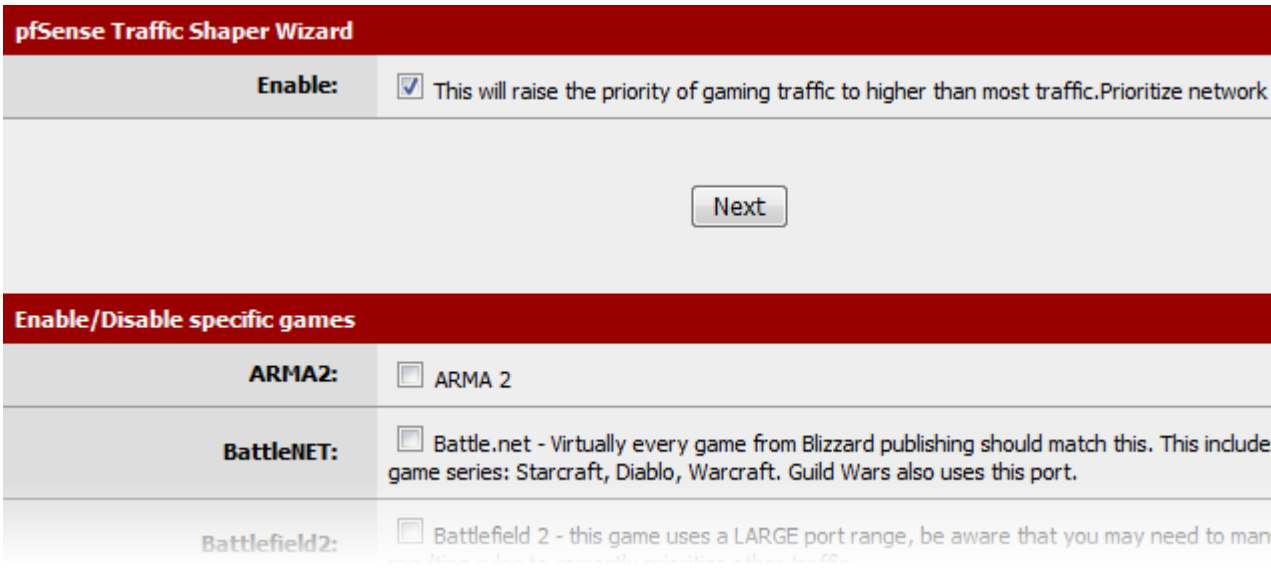
The remaining options are comprised of various known P2P protocols, more than 20 in all. Check each one that you would like to be recognized.

**Figure 21.5. Peer-to-Peer Networking**


# Network Games

Many games rely on low latency to deliver a good online gaming experience. If someone tries to download large files or game patches while playing, that traffic can easily swallow up the packets associated with the game itself and cause lag or disconnections. By checking the option to Prioritize network gaming traffic, as seen in Figure 21.6, “Network Games”, you can raise the priority of game traffic so that it will be transferred first and given a guaranteed chunk of bandwidth. There are many games listed, check all those which should be prioritized. If your game is not listed here you may still want to check a similar game so that you will have a reference rule that may be altered later.

**Figure 21.6. Network Games**

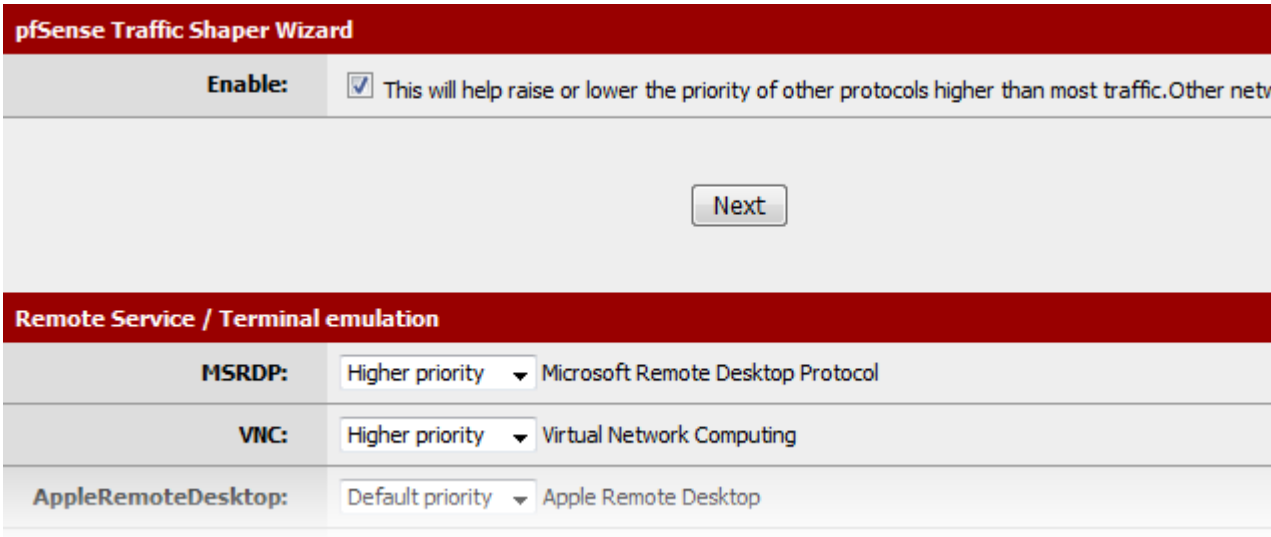


# Raising or Lowering Other Applications

The last configuration screen of the shaper wizard, seen in Figure 21.7, “Raise or Lower Other Applications”, lists many other commonly available application and protocols. How these protocols are handled will depend on the environment that this pfSense router will be protecting. Some of these may be desired, and others may not. For example, in a corporate environment, you may want to lower the priority of non-interactive traffic such as mail, where a slow down isn't noticed by anyone, and raise the priority of interactive services like RDP where poor performance is an impediment to people's ability to work. In a home, multimedia streaming may be more important, and other services can be lowered. Enable the option for Other networking protocols, and then pick and choose from the list.

There are more than 25 other protocols to choose from, and each can be given a **`Higher priority`**, **`Lower priority`**, or left at the **`Default priority`**. If you enabled p2pCatchAll, you will want to use this screen to ensure that these other protocols are recognized and treated normally, rather than penalized by the default p2pCatchAll rule.

**Figure 21.7. Raise or Lower Other Applications**




# Finishing the Wizard

All of the rules and queues will now be created, but not yet in use. By pressing the Finish button on the final screen, the rules will be loaded and active.

Shaping should now be activated for all new connections. Due to the stateful nature of the shaper, only new connections will have traffic shaping applied. In order for this to be fully active on all connections, you must clear the states. To do this, visit Diagnostics → States, click the Reset States tab, check Firewall state table, then click Reset.

# Shaper Wizard and IPv6

As of this writing, the shaper wizard did not create IPv6 rules, but will work if you create the rules manually or by cloning the existing rules.

# Monitoring the Queues

In order to be sure that traffic shaping is working as intended, it may be monitored by browsing to Status → Queues. As can be seen in Figure 21.8, “Basic WAN Queues”, this screen will show each queue listed by name, its current usage, and some other related statistics.

**Figure 21.8. Basic WAN Queues**

| Queue | Statistics | PPS | Bandwidth | Borrows | Suspends |
|---|---|---|---|---|---|
| **Interface WAN** | | | | | |
| Root queue | +/- | 12.0 | 96.66 Kbps | 0 | 0 |
| qACK | | 0.2 | 67 bps | 0 | 0 |
| qOthersDefault | | 1.2 | 7.71 Kbps | 0 | 0 |
| qP2P | | 2.2 | 8.10 Kbps | 0 | 0 |
| qVoIP | | 0.0 | 0 bps | 0 | 0 |
| qGames | | 0.0 | 0 bps | 0 | 0 |
| qOthersHigh | | 0.0 | 0 bps | 0 | 0 |
| qOthersLow | | 8.4 | 80.79 Kbps | 0 | 0 |

The graphical bar shows you how "full" a queue is. The rate of data in the queue is shown in both packets per second (pps) and bits per second (Kbps). Borrows happen when a neighboring queue is not full and capacity is borrowed from there when needed. Drops happen when traffic in a queue is dropped in favor of higher priority traffic. It is normal to see drops, and this does not mean that a full connection is dropped, just 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. 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.

# Advanced Customization

After using the shaper wizard, you may find that the rules it generates do not quite fit your needs. You may want to shape a service that is not handled by the wizard, a game that uses a different port, or there may be other services that need limited. Once the basic rules have been created by the wizard, it should be relatively easy to edit or copy those rules and create custom ones of your own.

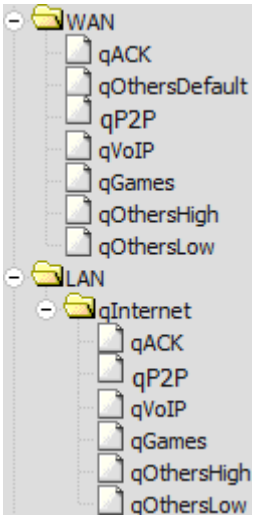
## Editing Shaper Queues

As mentioned in the summary, the queues are where bandwidth and priorities are actually allocated. Each queue will have settings specific to the scheduler that was chosen in the wizard, as mentioned earlier in the section called “ALTQ Scheduler Types”. The queues can also be assigned other attributes that control how they behave, such as being low-delay, or having certain congestion avoidance algorithms applied. Queues may be changed by going to Firewall → Traffic Shaper, and clicking on the queue names in the list or tree shown on the By Interface or By Queue tabs, like the one in Figure 21.9, “Traffic Shaper Queues List”

Editing queues is not for the weak of heart. It can be a complex task with powerful results, but without thorough understanding of the settings involved, it is best to stick with the queues generated by the wizard and alter their settings, rather than trying to make new ones from scratch.

To edit a queue, click its name in the list/tree. To delete a queue click it once to edit it, then click Delete This Queue. You should not attempt to delete a queue if it is still being referenced by a rule. To add a new queue, click the interface or parent queue under which the new queue will live, and then click Add New Queue.

**Figure 21.9. Traffic Shaper Queues List**




When editing a queue, each of the options should be carefully considered. If you are looking for more information about these settings than is mentioned here, visit the PF Packet Queueing and Prioritization FAQ [http://www.openbsd.org/faq/pf/queueing.html].[1]

The Queue Name must be between 1-15 characters and cannot contain spaces. The most common convention is to start the name of a queue with the letter "q" so that it may be more readily identified in the ruleset.

Priority 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 when there is an overload, so situate your queues accordingly. For example, VoIP traffic should be of the highest priority, so it should be set to a 7 on CBQ or 16 on PRIQ. Peer-to-peer network traffic, which can be delayed in favor of other protocols, should be set at 1.

The Queue Limit can set a packets per second limit on a queue, but is typically left blank. This is an alternate way to limit throughput.

There are four 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. Each interface should have one and only one default queue. |
| Random Early Detection (RED) | A method to avoid congestion on a link; it 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 the more bandwidth in use by a given connection, the more likely it is 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 can resend when needed. |
| Random Early Detection In and Out (RIO) | Enables RED with in/out, which will result in having queue averages being maintained and checked against for incoming and outgoing packets. |
| Explicit Congestion Notification (ECN) | Along with RED, it allows the sending of control messages that will throttle connections if both ends support ECN. Instead of |

[1] http://www.openbsd.org/faq/pf/queueing.html and also available in *The OpenBSD PF Packet Filter* book.

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.

The Description of the queue is up to you. It may be left blank, or filled with some text to explain the purpose of the queue.

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.

Next are the scheduler-specific options. They change depending on whether you have chosen HFSC, CBQ, or PRIQ. They are all described in the section called “ALTQ Scheduler Types”.

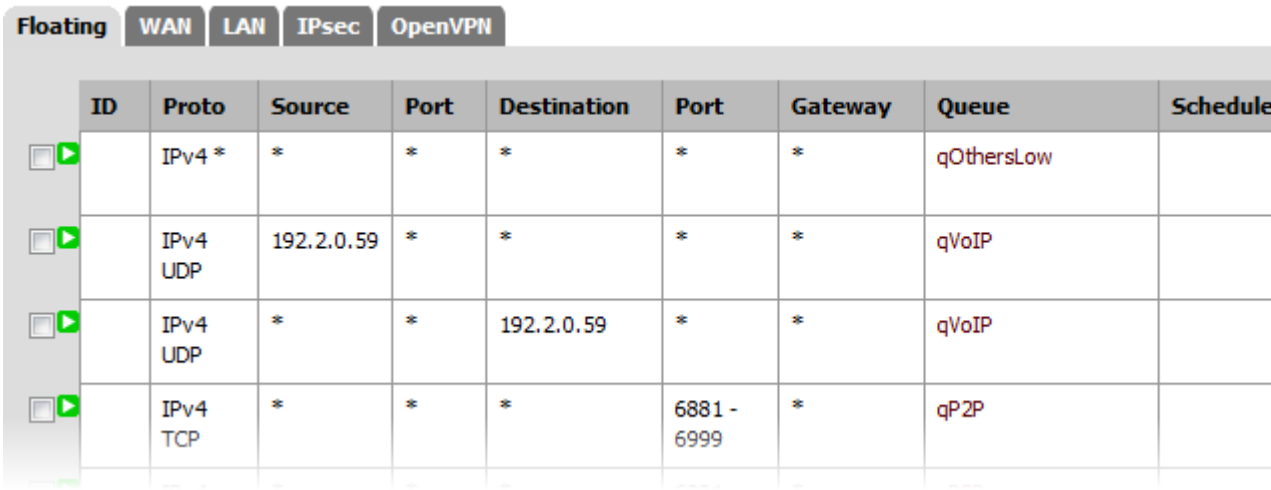
Click Save to save the queue settings and return to the queue list, then click Apply Changes to reload the queues and activate the changes.

# Editing Shaper Rules

Traffic shaping rules control how traffic is assigned into queues. If a packet matches a traffic shaper rule, it will be assigned into the queue specified by that rule. Packet matching is handled by firewall rules, notably on the Floating tab. To edit the shaper rules, go to Firewall → Rules, and click the Floating Tab. On that screen, shown in Figure 21.10, “Traffic Shaper Rules List”, the existing rules will be listed with the usual firewall rule attributes, including the Queues used by the rules. You may apply queues on rules on other tabs, but the wizard only makes rules on the Floating tab using the **`match`** action that does not affect the pass/block action, it only queues traffic. For more information on floating rules, see the section called “Floating Rules” for more information on floating rules, and the section called “Configuring firewall rules” for information on firewall rules in general. Because these rules are just like any other rules, you can match any way in which you are familiar.

On this screen also lies the master control for shaping. To remove the rules and queues created by the traffic shaper, and reset the shaper back to defaults, click Remove Shaper.

**Figure 21.10. Traffic Shaper Rules List**




# Limiters

Limiters are a new method of traffic shaping, introduced in pfSense 2.0 under Firewall → Traffic Shaper on the Limiters tab. Limiters use dummynet(4) [http://www.freebsd.org/cgi/man.cgi?

query=dummynet&apropos=0&sektion=0&manpath=FreeBSD+8.3-RELEASE&arch=default&format=html] to enact bandwidth limits and perform other prioritization tasks, among other things. Limiters are currently the only way to achieve per-IP bandwidth limiting in pfSense.

Limiters have actually been in use for a lot longer on pfSense as part of the Captive Portal's per-user bandwidth limits, but in 2.0 they have been hooked into pf so that they may be used on their own with normal firewall rules, outside of Captive Portal.

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 IP or the destination IP, so that the limits can be applied per-IP instead of as a group.

Limiters are almost always used in pairs, one for incoming traffic and one for outgoing traffic.

The `dummynet(4)` system was originally designed, according to its man page, 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 applying bandwidth limits for users or specific protocols, e.g. "Maximum of 1Mbit/s for SMTP", or "Joe's PC only can use 5bit/s". It can also apply a per-IP limit, such as "All Users in 192.168.50.0/24 can each use a maximum of 3Mbit/s each". Limiters are the only shaper type currently in pfSense capable of such oversubscription. The ALTQ shaper requires all child queues to sum up to no more than the speed of the parent queue, but masked limiters let you give a set limit to as many IPs as you can funnel through the limiter.

Another use for limiters is to apply a group limit for a chunk of your network, such as "All users in 192.168.40.0/24 can use a maximum of 5Mbit/s total" or "Do not let Bob the Bittorrent Guy use more than 2Mbit/s".

Limiters can also be used in a roundabout way to reserve bandwidth by limiting everything *except* a protocol you want to consume all bandwidth. In this type of setup on a 10Mbit/s link you'd pass traffic from, for example, a SIP server with no limiter, but then have a pass rule for all other traffic that gets a limit applied for 8Mbit/s. This would let the SIP server use all of the bandwidth it wanted, but it would always have at least 2Mbit/s, assuming the link speed is reliable/constant.

# How Limiters Work

Limiters, like ALTQ, hold traffic to a certain point by dropping or delaying packets to achieve a specific line rate. Usually a protocol's built-in mechanisms will detect the loss and back off to a speed it can sustain.

In situations where packets are queued under the same parent pipe, their weights are considered when ordering the packets before they are sent.

Unlike priorities in ALTQ's CBQ and PRIQ, the weight of a queue in a limiter will never cause it to starve for bandwidth.

# Limiters and IPv6

Limiters work with IPv6, though you will need separate IPv4 and IPv6 rules in order for the limiters to be applied properly.

# Limitations

Limiter pipes do not have a concept of borrowing bandwidth from other pipes. If you set a limit, it is a hard upper limit. Though `dummynet(4)` does support bursting, it does not support a rate for that, it only supports a given chunk of burst size if a queue is idle, and the pfSense GUI does not currently support setting that value.

Limiters use IPFW, so there will be some additional (though small) overhead from that kernel module being loaded and the extra packet processing involved.

Limiters cannot effectively guarantee a minimum bandwidth amount for a pipe or queue, only a maximum.

Queues cannot have bandwidth values, so you can't partition a pipe into smaller pipes, you can only use weights to prioritize packets inside a pipe.

The overhead from delaying and queueing packets can cause increased mbuf usage. For more information on increasing the amount of available mbufs, see the section called “Hardware Tuning and Troubleshooting”.

## Limiters and Multi-WAN

When using limiters with Multi-WAN, limits for non-default gateways will need to be applied using floating rules with the **`out`** direction and the appropriate gateway set.

# Creating Limiters

Limiters are managed under Firewall → Traffic Shaper on the Limiters tab. To create a new root-level limiter (pipe), click . To create a child limiter (queue), click the limiter under which it can be created, and click Add New Queue. You will nearly always want to use limiters in pairs at the same level (e.g. two pipes, or two queues), one for inbound traffic and one for outbound traffic, so keep that in mind when creating them.

Enable
: Check the box to enable this limiter. If the limiter is disabled, it will not be available as a choice to be used on 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 _.

  When choosing a name, it is best to 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. It is better to instead use *`Down`* or *`Download`*, and *`Up`* or *`Upload`*, but ultimately the naming convention can be anything you like, preferably something that makes sense to the firewall administrators.

Bandwidth (Pipes)
: This section lets you define 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`*.

  Burst
  : The Burst parameter specifies a total amount of data that can be transmitted without a limit applied after a period of idle time. This is not a rate, but a size. For example, if set to 2MB, then the user will transmit 2MB of data

at full speed, and once that bursting data size has been sent, the rate will be cut down to the limit specified in the Bandwidth field from that point until the limiter goes idle again. To disable bursting, enter a *0* here.

Bw Type — The units to be applied to the Bandwidth field, such as **Bit/s**, **Kbit/s**, **Mbit/s**, or **Gbit/s**.

Schedule — If you have defined schedules (the section called "Time Based Rules"), you may choose one here. When schedules are in use, you can define a bandwidth value for each potential schedule by clicking to add another bandwidth definition. If you use multiple bandwidth specifications, they must each have a different schedule defined. For instance if you have a "Work Day" schedule, you'll need an "Off Hours" schedule that contains all of the time not included in "Work Day".

Mask — This drop-down selection controls how the addresses in the limiter are masked. If left set to **none**, then no masking will be performed, and the pipe bandwidth will be applied to all traffic in the group as a whole.

If **Source Address** or **Destination Address** are chosen, then the pipe's bandwidth limit will be applied on a per-IP basis (or a subnet basis, depending on the masking bits), using the direction chosen in the masking. In general, you'll want to mask the **Source Address** on In (Upload) limiters for LAN-type interfaces, and **Destination Address** on Out (Download) 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 that is In on LAN would be used Out on WAN.

There are separate boxes to control the exact masking of addresses in the limiter. In pfSense 2.0.x, the IPv4 mask bits were always *32* fora Per-IPv4-address limit. This is the most common usage. For a per-IPv6-address limit, use 128 as the IPv6 mask bits value. If you wish to make per-subnet or similar masks, enter the subnet bits in the appropriate box for either IPv4 or IPv6 mask bits, such as *24* to limit in groups of /24 subnets.

Description — The Description, as usual, is an optional bit of text for your reference to explain the purpose for this Limiter.

Advanced Options — There are some additional options that vary whether you're editing a pipe or a queue, click Show Advanced Options and they will appear.

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. 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 to100. Higher values give more precedence to packets that are in a given queue. Unlike PRIQ and CBQ priorities, a lowly-weighted queue is not in danger of being starved of bandwidth.

| | |
|---|---|
| Packet loss rate | Another method of artificially degrading traffic, the Packet Loss Rate, can be configured to drop a certain fraction of the 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. As with the other fields, it is normally left empty so every packet is delivered. |
| Queue Size | Sets the size of the queue, specified in queue slots, used for handling queueing delay. Left blank, it defaults to 50, which is the recommended value. Slow speed links may need a lower queue size to operate efficiently. |
| 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. |

For more information about these values, you can also look at the ipfw(8) [http://www.freebsd.org/cgi/man.cgi?query=ipfw&sektion=8&apropos=0&manpath=FreeBSD+8.3-RELEASE] man page, in the section titled "Traffic Shaper (Dummynet) Configuration".

## Assigning and Using Limiters

Limiters are assigned using firewall rules, using the In/Out selectors in the advanced options section of the firewall rule. Any potential matching criteria you can express in a rule can be used to 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 single LAN single WAN setup, inbound traffic on a LAN interface is actually going toward the Internet, i.e. uploaded data. Outbound traffic on the LAN interface is going toward the client PC, i.e. downloaded data. When considering the WAN interface, the directionality is reversed; Inbound traffic is coming from the Internet to the client, and outbound traffic is going from the client to the Internet.

In most cases, both an In limiter and Out limiter will be selected, but you can choose to select only one if traffic should be limited in a single direction.

Limiters may be applied on normal interface rules, or on floating rules, even using the same **`Match`** action that can be used by ALTQ.

## 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. Each limiter's set bandwidth and parameters are displayed, along with the current traffic level moving inside the limiter. In the case of masked limiters, the bandwidth of each IP address is shown.

In the future, there will be an easier-to-read graphical representation of this limiter information. For now, having access to the raw information is useful until that particular feature has been completed.

# Layer 7 Inspection

Layer 7 inspection (L7) is, in simple terms, pattern matching to see if traffic matches a specific protocol such as HTTP, FTP, Bittorrent, etc. It compares packets against a given pattern that expresses how

traffic for a protocol should look, and if it finds a match, then it applies an action to the connection. In some places this is also known as Deep Packet Inspection (DPI). In pfSense, Layer 7 inspection can be used for traffic shaping using ALTQ or for blocking traffic.

Entries that contain patterns to match for Layer 7 inspection are called Layer 7 containers. These containers are maintained at Firewall → Traffic Shaper on the Layer 7 tab.

# Heavy CPU Requirements / Performance Penalty

Layer 7 inspection involves putting every packet that matches a specific rule through an inspection daemon. Not only is extra delay introduced because the packets are routed through a daemon, there is also overhead from running a pattern match on the payload of every packet involved. As such, depending on the amount of traffic you are inspecting, CPU used can *drastically* increase. Layer 7 *will* reduce your overall potential throughput. The only way around it is to use a more powerful CPU, but even that has its limits.

# Limitations

Layer 7 can help identify traffic, but there are some things it cannot do, such as:

- Matching encrypted traffic is not currently possible. If a user encrypts their traffic, such as Bittorrent, it can sidestep a layer 7 match.
- Some protocols vary too much to be reliably identified. Notable examples of this are Bittorrent and Skype. As aspects of the protocol change, new patterns may be needed.
- As mentioned previously, there is a heavy burden on the CPU to perform Layer 7 inspection.
- There is no affirmative case for protocol enforcement; HTTP traffic can be matched to be queued or blocked, but Layer 7 inspection cannot be used to ensure that *only* HTTP traffic is traveling on port 80.
- Layer 7 inspection cannot be used for routing or multi-wan decisions. By the time Layer 7 has identified a protocol, the connection has already been established, so it may not be re-routed along an alternate path.
- It is still necessary to pass traffic into Layer 7 inspection in order to for it to be processed. If a protocol hops between ports, such as Bittorrent, you would have to pass all traffic on all ports through Layer 7 inspection, at a great cost to your CPU, and if someone encrypts the protocol then it still can evade being matched.

# Layer 7 Patterns

The heart of how Layer 7 matching works are regular expression (regex) patterns that are compared to each packet queued up for inspection. pfSense ships with a stock set of patterns obtained from the L7-filter project [http://l7-filter.clearfoundation.com/], but new patterns may be created and uploaded (See the section called “Uploading New Patterns”).

# Creating Layer 7 Containers

To create a new Layer 7 container, go to Firewall → Traffic Shaper and click the Layer 7 tab. From there, click to add a new container.

| | |
|---|---|
| Enable/Disable | This checkbox controls whether or not the container is active and selectable on firewall rules. |
| Name | The name of the container, as it will appear for selection on firewall rules. The name must be alphanumeric, and may also include - and _. |

Description — An optional bit of text for your reference, to describe the purpose of this container.

Rules — The container may have one or more rules. Each pattern will be checked and the appropriate action applied to the packets flowing through the container. To add a new rule, click .

- Pattern — The name of file containing the match parameters for a given protocol or type of traffic, e.g. **`Bittorrent`**, **`smb`**, or **`http`**.
- Structure — The Structure option controls what type of behavior will occur once a pattern match has been found. The possible choices are: **`action`** for a firewall action, **`queue`** to place traffic into an ALTQ queue, and **`limiter`** to send traffic through a limiter.
- Behavior — This option controls the behavior that is dictated by the Structure of this rule. For an **`action`** structure, the only choice is to block. For **`queue`** structures, all available ALTQ queues are listed. For **`limiter`**, all enabled limiters are listed.

# Using Layer 7 Containers

Using a layer 7 container is a little counter-intuitive. As with other features, you enact them via firewall rules, however with a Layer 7 container, even if you want to block the traffic, you still use the pass action on the firewall rule.

The reason for this is due to how layer 7 inspection works. The rule passes the packet into the layer 7 container, and the container itself makes the decision to pass, block, queue, or limit the traffic based on whether or not a match was found.

To use a layer 7 container on a rule, first make sure it is a pass rule, and then choose the desired Layer 7 entry in the Layer 7 option of the rule, at the end of the Advanced Features section of the firewall rule edit screen.

# Uploading New Patterns

New patterns can be uploaded by clicking the link at the bottom of Firewall → Traffic Shaper on the Layer 7 tab. Before you can upload a pattern file, you have to first create one in the proper format, or download one from another source that is already in the proper format. For some examples, you can look in `/usr/local/share/protocols/`, the pfSense repository on Github, or the L7-filter project page linked earlier in this section.

Once you have crafted or located your custom pattern, visit the upload page, click browse, locate your pattern file, and select it. Once the file has been selected, click Upload Pattern File and the new pattern will be available for selection when editing a Layer 7 container.

# Traffic Shaping and VPNs

The following discussions pertain largely 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 of how the 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, it is more reliable to use methods of shaping that focus more on prioritization than bandwidth, such as PRIQ or in some cases, CBQ.

If the VPN contains *only* traffic that should be prioritized, then it is enough to consider only the VPN traffic itself on WAN, rather than attempting to queue traffic on the VPN as well as WAN. In these cases, you will need 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 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. Also, some features of OpenVPN can 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 some kind of 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 the section called “Interface assignment and configuration”, and then use the shaper wizard as if it were a separate WAN interface, and classify the traffic as needed.

## Shaping outside the tunnel (passtos)

If the primary concern is shaping VoIP traffic over a VPN, another choice to consider is OpenVPN's **passtos** option, called Type-of-Service in the pfSense OpenVPN options. This option will copy 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. The value can be matched using the DSCP option on firewall rules, as described in the section called “Diffserv Code Point”. Using this method, it is possible to prioritize VoIP traffic inside the VPN without actually treating the VPN as an additional WAN.

### Note

Because data from the inner packet is being copied to the outer packet, this does expose a little information about the type of traffic crossing the VPN. It is up to you to decide whether or not the information disclosure, though minor, is worth the risk for the gains offered by proper packet prioritization.

# IPsec

IPsec is presented to the operating system on a single interface no matter how many tunnels are configured and which are used by WANs the tunnels. This can pose some difficulties, 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. You can add your own floating rules to match and queue traffic on the IPsec interface, but you may find that only inbound traffic will be queued as expected, though actual results may vary.

# Troubleshooting Shaper Issues

Traffic Shaping/QoS is a tricky topic, and can prove difficult to get right the first time. There are some common pitfalls that people fall upon, which are covered in this section.

# Why isn't Bittorrent traffic going into the P2P queue?

Bittorrent is known for not using much in the way of standard ports. Clients are allowed to declare which port others should 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 traffic you want, and treat your default queue as low priority. You can use Layer 7 inspection to attempt to classify traffic, but it comes at a hefty CPU penalty and it still cannot classify encrypted protocols.

# Why isn't traffic to ports opened by UPnP properly queued?

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 you have using UPnP in your environment, this may be low priority traffic like Bittorrent, or high priority traffic like game consoles or voice chat programs like Skype. The queue can be set by going to Services → UPnP and entering a queue name into the Traffic Shaper Queue field.

That trick only works with ALTQ shaper queues, however. There is not currently a way to ensure that UPnP traffic gets a limiter applied properly, but we are hoping to have that addressed in a future release.

# How can I calculate how much bandwidth to allocate to the ACK queues?

This is a complex topic, and most people gloss over it and just guess a sufficiently high value. For more detailed explanations with mathematical formulas, check the Traffic Shaping section of the pfSense forums [http://forum.pfsense.org/index.php/board,26.0.html].[2] 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?

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. You may need to disable the WebGUI/ssh anti-lockout rules and perhaps even replace the default LAN to ANY firewall rule with more specific options. In the case of packages, you may need to adjust how your default queue is handled.

# My ISP changed my connection speed, but my shaper is still limiting my bandwidth to the old speed, how can I change it?

You need only edit the appropriate queues under Firewall → Traffic Shaper to obtain the new speed. The queues that need updating are the root queue for the WAN interface for the upload speed, and the qInternet queue on the LAN or other internal interface for the download speed. If you have multiple WANs, the qInternet queue will be the summed download speed of all WANs, so adjust it by the amount that your ISP download bandwidth changed.

Alternately, if the wizard was used, and no custom changes to the queues or rules were made, then the wizard can be run again, and the value can be updated in the wizard.

[2] http://forum.pfsense.org/index.php/board,26.0.html

# Chapter 22. Server Load Balancing

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 Chapter 15, *Multiple WAN Connections*. Server load balancing allows you to distribute traffic to multiple internal servers for load distribution and redundancy, and is the subject of this chapter.

Server load balancing allows you to distribute traffic between multiple internal servers. It is most commonly used with web servers and SMTP servers though can be used for any service that uses TCP, 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 configuration. For large or complex deployments, you will commonly want a more powerful solution. However for basic needs, the functionality available in pfSense suits countless sites very well. There are some more full-featured load balancer packages available for pfSense also, such as HAProxy and Varnish, but the built-in load balancer based on OpenBSD's **relayd** does a great job for many deployments. It has better monitoring options than those which existed in pfSense 1.2.x. Now it 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.

# Explanation of Configuration Options

There are two portions of configuration for the server load balancer. Pools define the list of servers to be used, which port they listen on, and the monitoring method to be used. Virtual Servers define the IP and port to listen on, and the appropriate pool to direct the incoming traffic destined to that IP and port. Monitors are used to create custom monitoring methods, and Settings contains some global options that alter how the load balancer operates.

## Pools

To configure Pools, browse to Services → Load Balancer on the Pools tab. Click to add a new pool. Each of the options on this page is discussed here.

| | |
|---|---|
| Name | Enter a name for the pool here. The name is how the pool is referenced later 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 separator is an underscore. |
| Mode | Select **`Load Balance`** to balance load between all the servers in the pool, or **`Manual Failover`** to always use the servers in the Enabled list, and you can manually move them between an enabled and disabled state. |
| Description | Optionally enter a longer description for the pool here. |
| Port | This is the port your servers are listening on internally. This can be different from the external port, which is defined later in the virtual server configuration. You can use an alias to define multiple ports, however, if you do so then you 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 balancer determines if the servers are up. 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 pinging them, and will mark them down if they do not respond to pings. There are many more types of monitors, and they can be customized. They will be covered in more detail later in the chapter.

Server IP Address — This is where you fill in the internal IP address of the servers in the pool. Enter them one at a time, clicking Add to pool afterwards.

Current Pool Members — This field shows the list of servers you have added to this pool. You can remove a server from the pool by clicking on its IP address and clicking Remove. There are two lists in this section, Pool Disabled, and Enabled (default). The servers in the Enabled (default) list are active an 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 >.

After populating all the fields as desired, click Save. Proceed to configuring the Virtual Server for this pool by clicking the Virtual Servers tab.

If you want to do automatic failover, you will need to create a second pool to be used as a Fall Back Pool, so create another new pool and add your secondary group of servers.

## Virtual Servers

The Virtual Servers tab under Services → Load Balancer is where you define the IP and port to listen on for forwarding traffic to the previously configured pools. Click to add a new virtual server. Each of the options on this page will be discussed here.

Name — Enter a name for the virtual server here. This is for your reference, but must also adhere to the same limits as an alias or interface name. Letters and numbers only, the only separator is an underscore. No spaces or slashes.

Description — Optionally enter a longer description for the virtual server here. This is also just for reference purposes, and does not have any formatting limits.

IP Address — This is where you enter the IP address upon which the virtual server will listen. This is usually your WAN IP or a Virtual IP on WAN. It must be a static IP address. You can use a CARP VIP here for a high availability load balancer setup. For more information on high availability and CARP VIPs, refer to Chapter 25, *Firewall Redundancy / High Availability*. You may also use an IP Alias VIP, or a Proxy ARP VIP.

Furthermore, you may also use an Alias here to specify multiple IP addresses upon which this virtual server may accept connections.

Port — This is the port upon which the virtual server will listen. It can be different from the port your servers are listening on internally. You can use an alias to define multiple ports, however, if you do so then you must use the same port alias here and in the Pool configuration.

Virtual Server Pool — This is where you select the previously configured pool. The connections to the IP Address and Port defined on this screen will be directed to the IPs and port configured in the pool.

Fall Back Pool — This is the alternate pool that clients are directed to if all the servers in your primary pool are down. If you don't have an alternate server to send requests to, you may 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, you can setup a simple web server to return a basic maintenance page for any request and use it as your 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 in as though they were hitting a traditional NAT rule. Your servers will see the original source IP of the client, no proxying is performed.
- In **DNS** mode, **relayd** acts as a DNS proxy. It will balance the load over multiple DNS servers, but the original client IP is lost, they will see the firewall as the source of the DNS query. Keep this in mind when setting up any kind of views or source-based queries/ restrictions on DNS servers involved in load balancing.

After filling in the fields appropriately, click Submit, then Apply Changes.

# Monitors

Monitors are configured on the Monitors tab under Services → Load Balancer. There are five basic pre-defined Monitor types (ICMP, TCP, HTTP, HTTPS, and SMTP), but you can add as many custom types as you like as well to better detect your own specific types of failures.

## Pre-defined Monitors

The pre-defined monitors are included in the default configuration. They consist of:

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 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 it gets a 200 response code, 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 the included monitors are not good enough for your purposes, or you need to tweak them slightly, then you can create your own custom monitor. Most monitor types have their own specific settings that can be customized to suit your needs. Click on the Monitors tab to add a new monitor.

First, give your monitor a Name. This must adhere to the same limits as an alias or interface name. Letters and numbers only, the only separator is an underscore. No spaces or slashes. Next, a Description can be added to give a more thorough explanation about the purpose of this monitor.

The remaining options vary based on the Type you select.

ICMP & TCP
: These have no extra options. Any custom monitor using these types will behave identically to the pre-defined monitor of the same name. There shouldn't be any need to create a custom version of these, unless the original was deleted.

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 you may define to control the behavior of the monitor.

  Path
  : The Path defines the path section of the URL sent to the server. If your site contains mostly dynamic content, or the base URL does a redirect, it is probably 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 your server runs multiple virtual hosts, you can use this field to define which hostname is sent with the request so that the expected response can be achieved.

  HTTP Code
  : This defines the expected response you want from the server, given the request to the Host/Path. Most commonly this would be set to **`200 OK`**, but if your server uses another return code that would be expected as a healthy response to this query, you can choose it here. If you are unsure of the return code you want, you can inspect your 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 you can define 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.

Once you have defined the monitor to your liking, press Save.

## Settings

In addition to the per-pool or per-server options, there are also some global options that control the behavior of relayd. These settings are under Services → Load Balancer on the Settings tab. These are:

Timeout
: Set the global timeout in milliseconds for checks. Leave blank to use the default value of 1000 ms (1 second). If you have a loaded server pool that takes longer to respond to requests, you can increase this timeout.

Interval
: Set the interval in seconds at which the member of a pool will be checked. Leave blank to use the default interval of 10 seconds. If you need 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 you have a busy server, you can increase this amount to accommodate the load.

## Firewall rules

The last step is to configure firewall rules to allow traffic to the pool. Just like in a NAT scenario, the firewall rules must permit traffic to the internal private IPs of the servers, as well as the port they

are listening on internally. You should create an alias for the servers in the pool, 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 destination of the alias created for the pool. A specific example of this is provided in the section called “Configuring firewall rules”. For more information on firewall rules, refer to Chapter 10, *Firewall*.

## Sticky connections

There is one additional configuration option available for server load balancing, under the System → Advanced menu, on the Miscellaneous tab. Under Load Balancing, you will find Use sticky connections. Checking this box will ensure clients with an active connection to the pool are always directed 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 a particular client's requests should only go to a single server, for session or other reasons. Note this isn't perfect, as if the client's web browser closes all TCP connections to your 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 you are strictly reliant on a specific client never getting a different server in the pool regardless of how long the browser sits there inactive, you should 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.

# Web Server Load Balancing Example Configuration

This section shows you how to configure the load balancer from start to finish for a two web server load balanced environment.

## Example network environment

**Figure 22.1. Server load balancing example network**

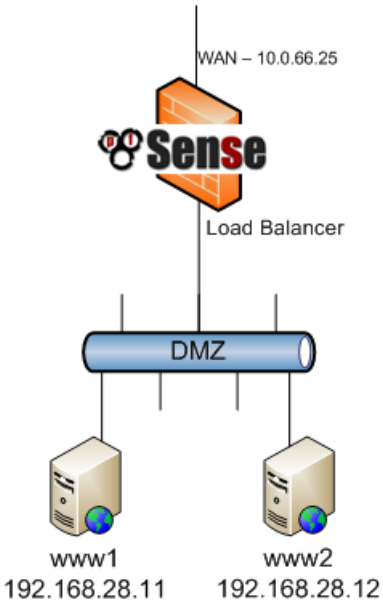
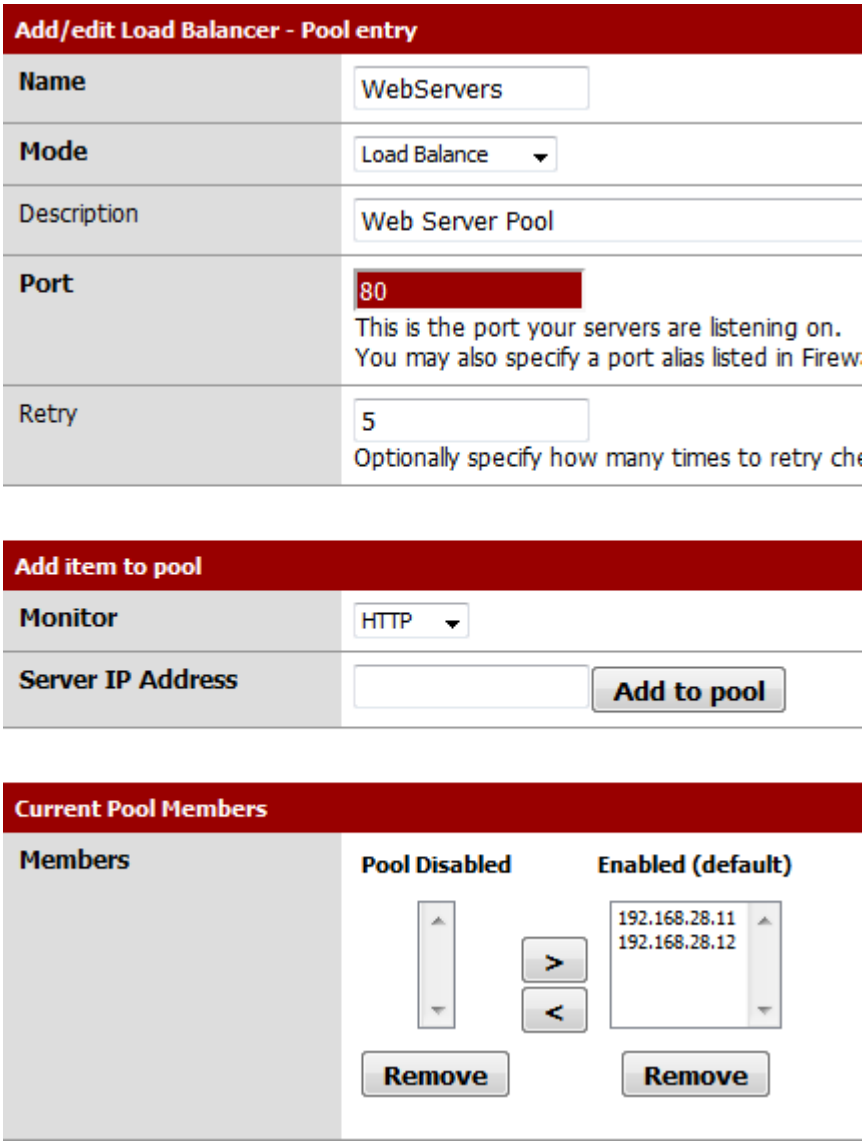



Figure 22.1, “Server load balancing example network” shows the example environment configured in this section. It consists of a single firewall, using its WAN IP for the pool, with two web servers on a DMZ segment.

# Configuring pool

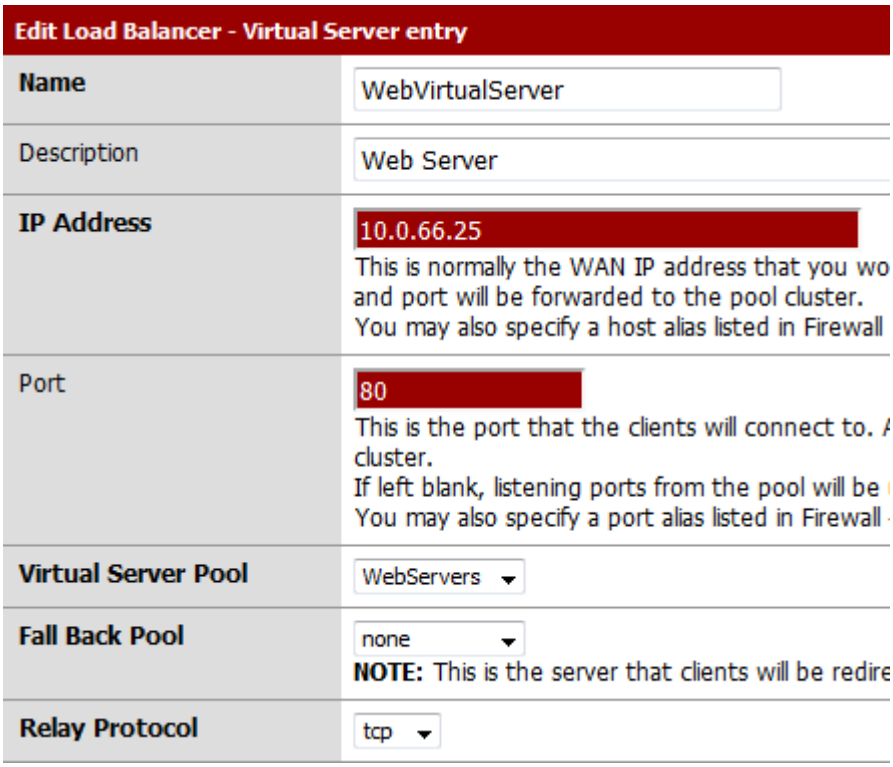
To configure the pool, browse to Services → Load Balancer on the Pools tab and click . Figure 22.2, “Pool configuration” shows the load balancing pool configuration for the two web servers, using an HTTP monitor. After filling in all the fields appropriately, click Save.

### Figure 22.2. Pool configuration



# Configuring virtual server

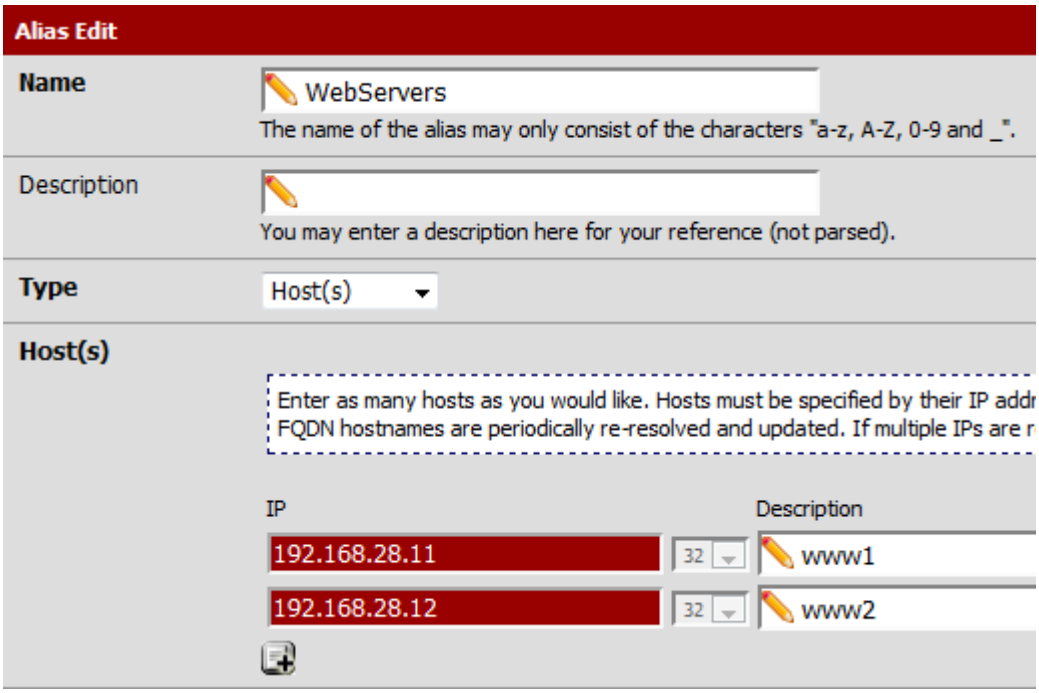
**Figure 22.3. Virtual Server configuration**




On the Virtual Servers tab, click to add a new virtual server. Figure 22.3, "Virtual Server configuration" shows the virtual server configuration to listen on the WAN IP (10.0.66.25) on port 80 and forward the traffic on that IP and port to the servers defined in the **WebServers** pool. For the Fall Back Pool, this configuration uses **none** because of lack of another option. In this case, if both of the pool servers are down, the virtual server is inaccessible. After filling in the fields here, click Submit, then Apply Changes.

# Configuring firewall rules

**Figure 22.4. Alias for web servers**




Now 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. It is preferable to use an alias containing all the servers in the pool, so access can be allowed with a single firewall rule. Browse to Firewall → Aliases and click to add an alias. Figure 22.4, “Alias for web servers” shows the alias used for this example configuration, containing the two web servers.

Click Save after entering the alias, and Apply Changes. Then browse to Firewall → Rules and on the tab for the interface where the client traffic will be initiated (WAN in this case), click . Figure 22.5, “Adding firewall rule for web servers” shows a snippet of the firewall rule added for this configuration. The options not shown were left at their defaults, aside from Description.

**Figure 22.5. Adding firewall rule for web servers**

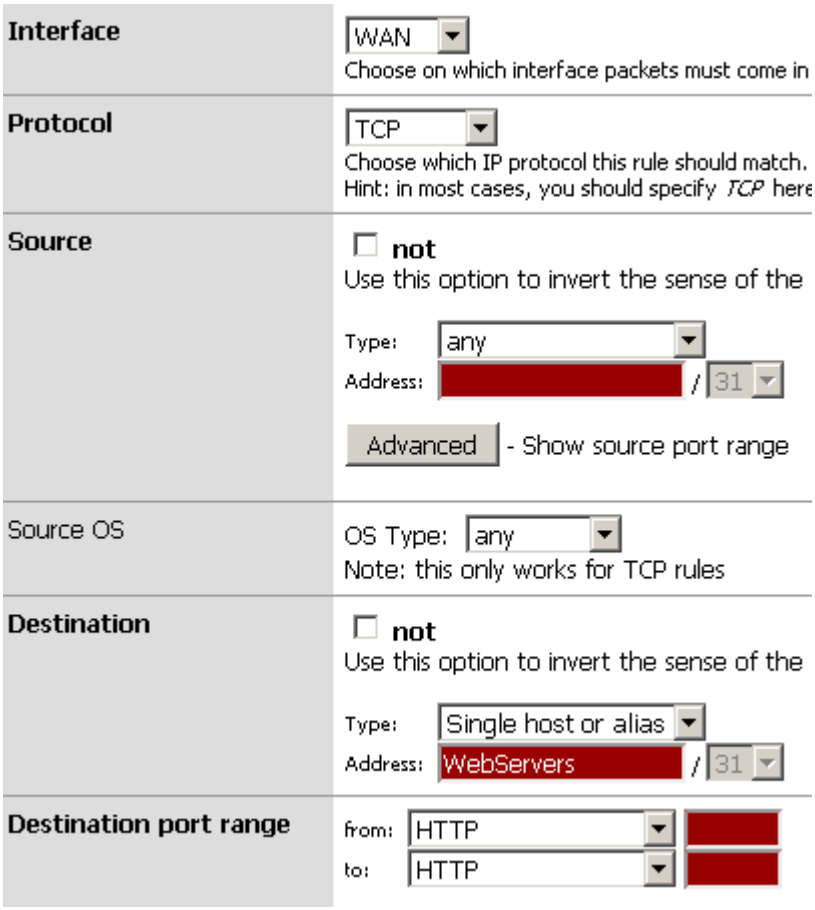



Figure 22.6, “Firewall rule for web servers” shows the rule after it was added.

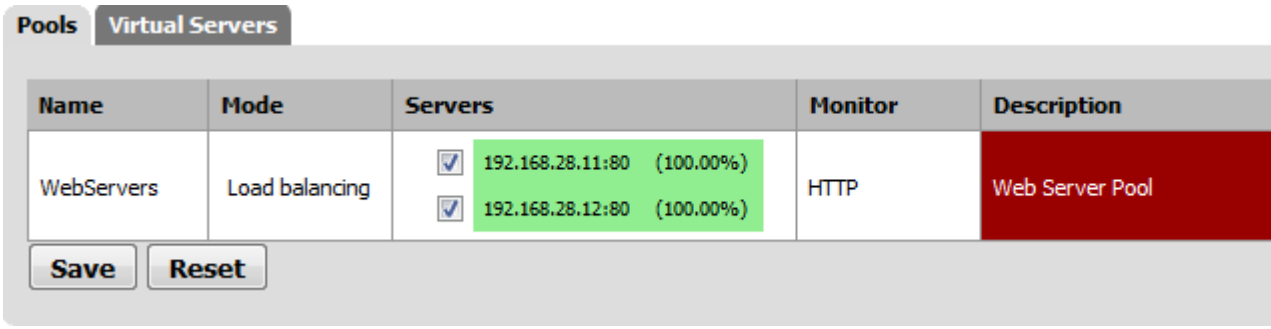
**Figure 22.6. Firewall rule for web servers**


# Viewing load balancer status

Now that your load balancer is configured, to view its status, browse to Status → Load Balancer and click the Virtual Servers tab. Here you will see the status of the server as a whole, typically listed as either Active or Down.

On the Pools tab you will see an individual status for each member of a Pool (as shown in Figure 22.7, “Pool status”). You will see the row for a server in green if it is online, and in 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 makes them disabled in the pool, the same as moving them between the enabled and disabled list on the pool editing page. If you wish to disable a server, uncheck it, then press Save.

**Figure 22.7. Pool status**




If you stop the web server service on one of the servers or take the server off the network entirely if using ICMP monitors, you will see the status update to Offline and the server will be removed from the pool.

## Verifying load balancing

To verify the load balancing, **curl** is the best option to ensure your web browser's cache and persistent connections do not affect the results of your testing. **curl** is available for every OS imaginable and can be downloaded from the curl website [http://curl.haxx.se]. To use it, simply run **curl http://*mysite*** replacing *mysite* with either the IP address or hostname of your site. You must do this from outside your network. The following illustrates an example of testing with **curl** from the WAN side.

```
# curl http://10.0.66.25
This is server www2 - 192.168.28.12
# curl http://10.0.66.25
This is server www1 - 192.168.28.11
```

When initially testing your load balancing, you will want to configure each server to return a page specifying its hostname, IP address, or both, so you will know which server you are hitting. If you do not have sticky connections enabled, you will get a different server each time you request a page with **curl**.

# Troubleshooting Server Load Balancing

This section describes the most common issues users encounter with server load balancing, and how to troubleshoot them.

## 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 just re-uses the existing connection. A single connection will never be changed to another balanced server. Another common issue is the cache of your web browser, where the browser never actually requests the page again. It's preferable to use a command line tool such as **curl** for testing of this nature, because it ensures you are never impacted by the problems inherent in testing with web browsers — it has no cache, and opens a new connection to the server each time it is run. More information on **curl** can be found in the section called “Verifying load balancing”.

If you are using sticky connections, ensure you are testing from multiple source IPs. Tests from a single source IP will always go to a single server unless you wait long times in between connections.

## Down server not marked as offline

If a server goes down but is not marked as offline, it's because from the perspective of the monitoring that pfSense is doing, it isn't really down. If using a TCP monitor, that TCP port is 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 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's because it isn't online from the perspective of the firewall. The server must answer on the TCP port used or respond to pings sourced from the interface IP 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 firewall's LAN IP. To verify this for ICMP monitors, browse to Diagnostics → Ping and ping the server IP using the interface where the server is located. For TCP monitors, log into the firewall using SSH, or at the console, and choose console menu option **8**. At the command prompt, try to telnet to the port where the server is listening. For example to test a web server in the example earlier in this chapter, you would run **`telnet 192.168.28.11 80`**.

A failed connection will sit there for a while trying to connect, while a successful connection will connect immediately. The following is an example of a failed connection.

```
# telnet 192.168.28.12 80
Trying 192.168.28.12...
telnet: connect to address 192.168.28.12: Operation timed out
telnet: Unable to connect to remote host
```

And here is an example of a successful connection.

```
# telnet 192.168.28.12 80
Trying 192.168.28.12...
Connected to 192.168.28.12.
Escape character is '^]'.
```

You will likely find that the connection fails, and will need to troubleshoot further on the server.

# Unable to reach a virtual server from a client in the same subnet as the pool server

If you find that you are unable to reach a server from a client PC inside the same subnet as the actual pool (backend) server, the typical problem is that relayd forwards the connection in 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 and not the IP address in relayd. Then, due to the fact that the server IP is incorrect from the view 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 your internal interface (LAN) coming from the LAN subnet, going to the server, 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.

# Chapter 23. Wireless

pfSense includes built in wireless capabilities that allow you to turn your pfSense install into a wireless access point, use a wireless 802.11 connection as a WAN connection, or both. This chapter also covers suggested means of securely accommodating external wireless access points, and how to securely deploy a wireless hotspot. In-depth coverage of 802.11 is outside the scope of this book. For those seeking such information, we recommend the book *802.11 Wireless Networks: The Definitive Guide* [http://www.amazon.com/gp/product/0596100523?ie=UTF8&tag=pfsense-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0596100523].

The most significant change in Wireless support from 1.2.3 to 2.x versions is the new support for Virtual Access Points (VAPs) on supported hardware. This feature allows you to have multiple access points or stations on a single card. They all use the same channel but have unique interfaces, subnets, security settings, etc. This allows you to have, for example, one secure wireless network with greater access than a less secure guest wireless network.

# Recommended Wireless Hardware

There are a variety of wireless cards supported in FreeBSD 8.3, and pfSense includes support for every card supported by FreeBSD. Some are supported better 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 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 chipsets 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 you have no way of knowing which "minor" card revision you will end up obtaining. 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 you already have one, it's worth trying to see if it is compatible, but be warned if you purchase one because the "same" model worked for someone else, you may end up with a completely different piece of hardware that is incompatible.

## Status of 802.11n Support

pfSense 2.1 is based on FreeBSD 8.3, which still lacks some of the required functionality for 802.11n support. However, some cards are supported that work on 802.11n frequencies, specifically newer Atheros cards, `mwl(4)` cards, and some `run(4)` cards, among others. These may work using the 802.11n standard but not at 802.11n speeds. In pfSense 2.2 we will be using a FreeBSD 10.x base which should bring along 802.11n support at that time.

## Wireless drivers included in 2.1

This section lists the wireless drivers included in pfSense 2.1, and the chipsets that are supported by those drivers (pulling from the FreeBSD man pages for the drivers). Drivers in FreeBSD are referred to by their driver name, followed by (4), such as `ath(4)`. The (4) refers to kernel interfaces, in this case specifying a network driver. The drivers are listed in order of frequency of use with pfSense, based on mailing list and forum postings since the project's inception.

For more detailed information on cards supported, and the most up to date information, refer to the pfSense wiki [http://doc.pfsense.org/index.php/Supported_Wireless_Cards].

# Cards Supporting Access Point (hostap) Mode

The cards in this section support acting as an access point to take connections from other wireless clients. This is referred to as **hostap** mode.

## ath(4)

Supports cards based on the Atheros AR5210, AR5211, AR5212 and AR5416 APIs which are used by many other Atheros chips of varying model numbers. We have included the `ath(4)` drivers from FreeBSD 9.x, so it also supports some AR92xx and similar cards. Most Atheros cards support up to four virtual access points (VAPs) or stations (or a combination to create a wireless repeater).

## ral(4) / ural(4) / rum(4) / run(4)

Ralink Technology IEEE 802.11 wireless network drivers. `ral(4)` supports cards based on the Ralink Technology RT2500, RT2501 and RT2600 chipsets. `ural(4)` supports RT2500USB. `run(4)` supports RT2700U, RT2800U and RT3000U and similar. `rum(4)` supports RT2501USB and RT2601USB and similar. Of these, only certain chips supported by `run(4)` can support VAPs.

## mwl(4)

Marvell IEEE 802.11 wireless network driver — supports cards based on the 88W8363 chipset. This card supports multiple VAPs and stations, up to eight of each.

## wi(4)

Lucent Hermes, Intersil PRISM and Spectrum24 IEEE 802.11 driver — supports cards based on Lucent Hermes, Intersil PRISM-II, Intersil PRISM-2.5, Intersil Prism-3, and Symbol Spectrum24 chipsets. These cards support only 802.11b.

## an(4)

Aironet Communications 4500/4800 wireless network adapter driver — supports Aironet Communications 4500 and 4800 wireless network adapters and variants.

# 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.

## uath(4)

Atheros USB 2.0 wireless devices using AR5005UG and AR5005UX chipsets.

## ipw(4) / iwi(4) / iwn(4) / wpi(4)

Intel wireless network drivers for various models. `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.

Several of the Intel adapters also come with a license restriction that you may have noticed in the boot log. The `ipw(4)`, `iwi(4)`, and `wpi(4)` drivers have license files that must be read 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. If you agree to the license, you must then edit `/boot/loader.conf.local` and add a line to indicate the license acknowledgement, such as:

```
legal.intel_ipw.license_ack=1
```

Given the limited use of these adapters (client mode only), a GUI-based solution to acknowledging these licenses has not yet been created.

### bwi(4) / bwn(4)

Broadcom BCM43xx IEEE 802.11b/g wireless driver.

### malo(4)

Marvell Libertas IEEE 802.11b/g wireless driver.

### upgt(4)

Conexant/Intersil PrismGT SoftMAC USB IEEE 802.11b/g wireless driver.

### urtw(4)

Realtek RTL8187B/L USB IEEE 802.11b/g wireless network driver.

### zyd(4)

ZyDAS ZD1211/ZD1211B USB IEEE 802.11b/g wireless network device.

## 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 [https://docs.google.com/spreadsheet/ccc?key=0AojFUXcbH0ROdHgwYkFHbkRUdV9hVWljVWl5SXkxbFE] linked from the documentation wiki article on wireless support [http://doc.pfsense.org/index.php/Supported_Wireless_Cards]. 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.

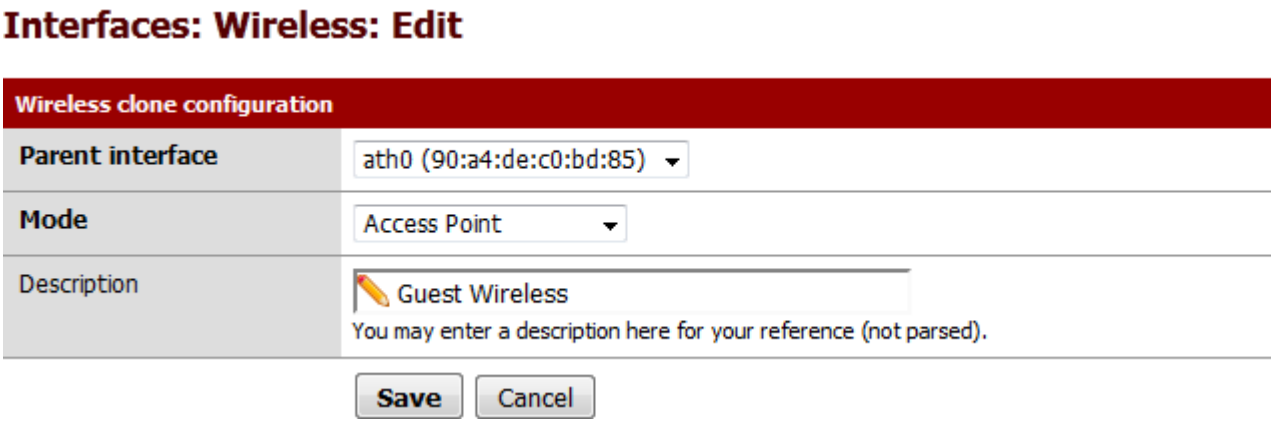
# Working with Virtual Access Point Wireless Interfaces

Starting with pfSense 2.0, the concept of virtual wireless interfaces was introduced. These are referred to as Virtual Access Point or VAP interfaces, even if they're being used for client mode. As mentioned at the start of this chapter, VAPs allow you to run multiple access points on the same wireless card, or to use them as a combination of access point and client mode. Support for VAPs varies by card and driver, so be sure to check the information on driver support in the section called “Recommended Wireless Hardware”. Odds are, however, if you have an Atheros wireless card it will work.

If you only plan on using a single wireless interface and no VAPs, then you need not worry about these settings. By selecting the physical interface the firewall will automatically create one for you behind the scenes so you do not need to manually make one.

To manage VAPs, navigate to Interfaces → (assign) on the Wireless tab. To add a new VAP, click . From that screen, you select the Parent Interface, pick the Mode from one of **`Access Point`**, **`Infrastructure`** (BSS, client mode), or **`Ad-hoc`** (IBSS), and give the VAP a Description. An example is shown in Figure 23.1, “Adding a VAP”.

**Figure 23.1. Adding a VAP**




Once you save the VAP, it is then available for assignment under Interfaces → (assign). From there, you edit the settings like any other wireless interface. Keep in mind that you must configure the resulting interface to be the mode selected here.

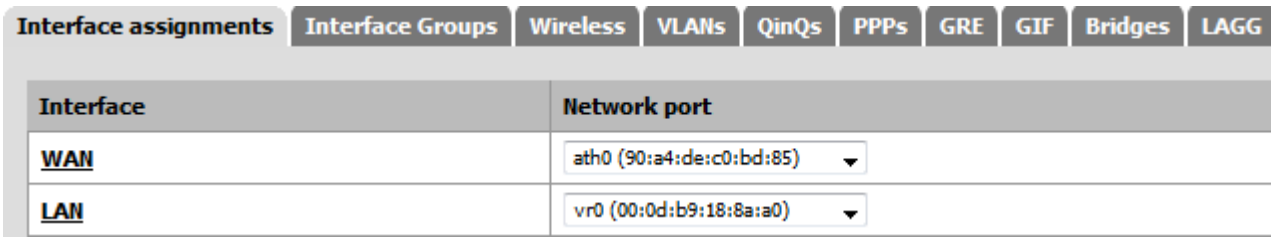
# Wireless WAN

You can assign your wireless card as your WAN interface, or an OPT WAN in a multi-WAN deployment. This section covers assigning and configuring a wireless interface as a WAN interface.

## Interface assignment

If you have not already assigned your wireless interface, browse to Interfaces → (assign). Click Add to add an OPT interface for your wireless, or select it as WAN if desired. Figure 23.2, “Interface assignment — wireless WAN ” shows an Atheros card assigned as WAN.

**Figure 23.2. Interface assignment — wireless WAN**




## Configuring your wireless network

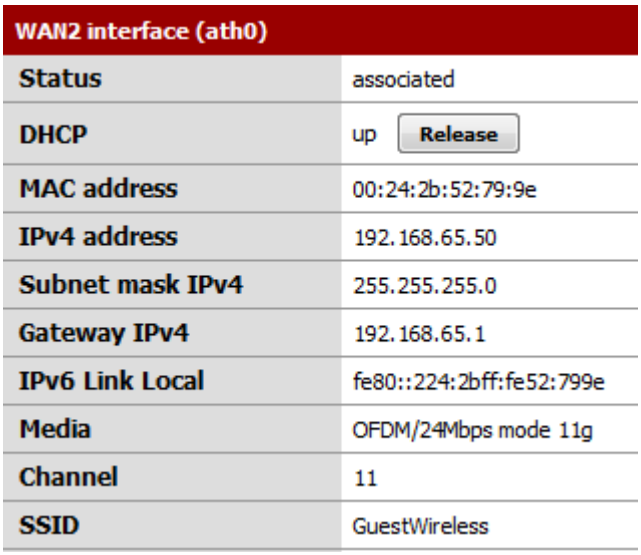
Browse to the Interfaces menu for your wireless WAN interface. This example uses WAN, so I will browse to Interfaces → WAN. Select the type of configuration (DHCP, static IP, etc.), and scroll down under Wireless configuration. Make sure the Standard is set appropriately, for example **`802.11g`**. Choose Infrastructure (BSS) mode, fill in the SSID, and configure encryption such as WEP (Wired Equivalent Privacy) or WPA (Wi-Fi Protected Access) if used. Most wireless networks will not need any further configuration, but if yours does, make sure it's configured appropriate for the access point you will be using. Then click Save.

## Checking wireless status

Browse to Status → Interfaces to see the status of the wireless interface just configured. You can tell whether the interface has successfully associated with the chosen access point by looking at the

status of the interface. Status associated means it is connected successfully, as shown in Figure 23.3, “Wireless WAN Associated”.

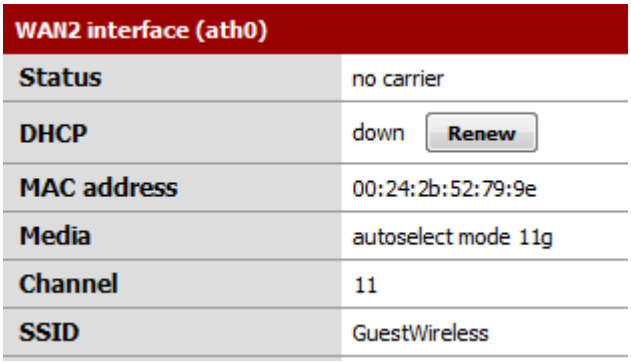
### Figure 23.3. Wireless WAN Associated



| WAN2 interface (ath0) | |
|---|---|
| Status | associated |
| DHCP | up Release |
| MAC address | 00:24:2b:52:79:9e |
| IPv4 address | 192.168.65.50 |
| Subnet mask IPv4 | 255.255.255.0 |
| Gateway IPv4 | 192.168.65.1 |
| IPv6 Link Local | fe80::224:2bff:fe52:799e |
| Media | OFDM/24Mbps mode 11g |
| Channel | 11 |
| SSID | GuestWireless |

If it shows No carrier, it was unable to associate. Figure 23.4, “No carrier on wireless WAN” shows an example of this, where I disconnected the antenna so it could not connect to a wireless network that was a few rooms away.

### Figure 23.4. No carrier on wireless WAN

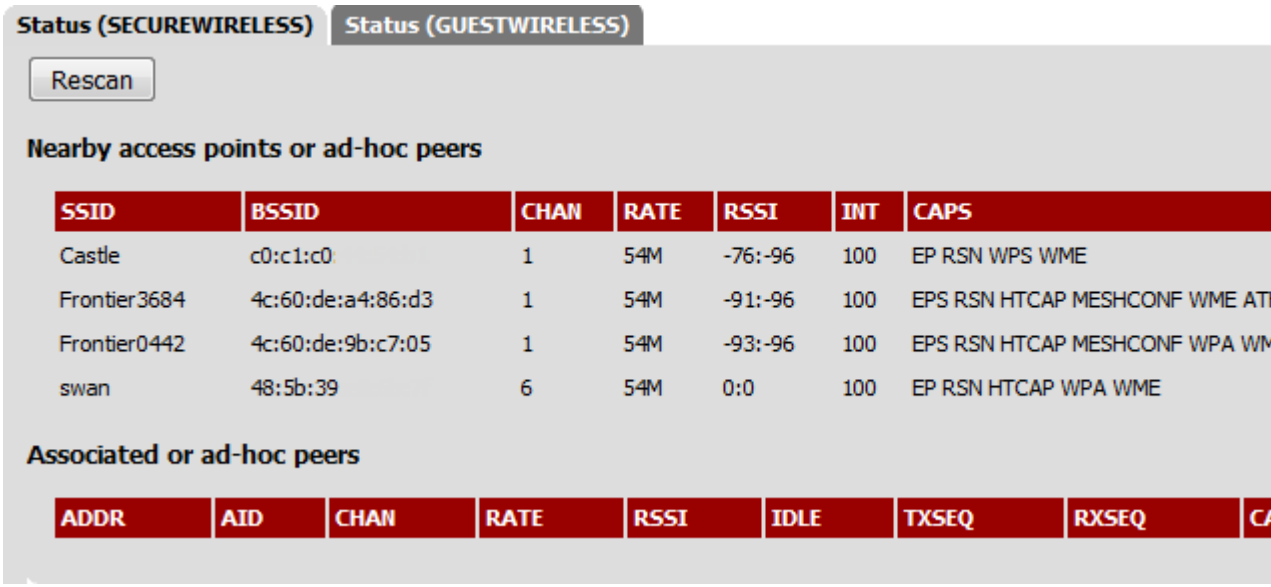


| WAN2 interface (ath0) | |
|---|---|
| Status | no carrier |
| DHCP | down Renew |
| MAC address | 00:24:2b:52:79:9e |
| Media | autoselect mode 11g |
| Channel | 11 |
| SSID | GuestWireless |

# Showing available wireless networks and signal strength

By browsing to Status → Wireless, you can see the wireless networks visible to your firewall as shown in Figure 23.5, “Wireless Status”. Your wireless interface must be configured before this menu item will appear.

**Figure 23.5. Wireless Status**




# Bridging and wireless

Only wireless interfaces in access point (hostap) mode will function in a bridged configuration. You can bridge a wireless interface in hostap to any other interface to combine the two interfaces on the same broadcast domain. You may wish to do this if you have devices or applications that must reside on the same broadcast domain to function properly. This is discussed in more depth in the section called “Choosing bridging or routing”.

## BSS and IBSS wireless and bridging

Because of the way wireless works in BSS (Basic Service Set) and IBSS (Independent Basic Service Set) mode, and the way bridging works, you cannot bridge a wireless interface 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 desirable — it is just 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 [http://lists.freebsd.org/pipermail/freebsd-current/2005-October/056977.html].[1] 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 simply 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 2.0.

# Using an External Access Point

If you have an existing wireless access point, or a wireless router that you wish to only use as an access point now that pfSense is acting as your firewall, there are several ways to accommodate wireless in your network. This section covers the most commonly deployed scenarios. This type of deployment is popular for wireless because it makes it 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 you can have an 802.11n wireless network and have it still be secured by pfSense, even though we do not yet have support for 802.11n devices.

[1] http://lists.freebsd.org/pipermail/freebsd-current/2005-October/0 56977.html

# Turning your wireless router into an access point

When replacing a simple wireless router such as a Linksys or D-Link or other home grade device with pfSense as a perimeter firewall, the wireless functionality can be retained by turning the wireless router into a wireless access point by following the steps described in this section. These are generic steps that need to be followed for any device. To find specifics for your wireless router, refer to its documentation.

## Disable the DHCP server

First you will want to disable the DHCP server if it was previously in use. You will want pfSense to handle this function for your network, and having two DHCP servers on your network will cause problems.

## Change the LAN IP

Next you will need to change the LAN IP to an unused IP on the subnet where your access point will reside (commonly LAN). It is probably using the same IP you will assign to the pfSense LAN interface, so it will require a different address. You will want to retain a functional IP on the access point for management purposes.

## Plug in the LAN interface

Most wireless routers bridge the wireless onto the internal LAN port or ports. This means the wireless will be on the same broadcast domain and IP subnet as the wired ports. For routers with an integrated switch, any of the switch ports will usually work.

### Note

*You do not want to plug in the WAN or Internet port on your router!* This will put your wireless network on a different broadcast domain from the rest of your network, and will result in NATing traffic between your wireless and LAN and double NATing traffic between your wireless and the Internet. This is an ugly design, and will lead to problems in some circumstances, especially if you need to communicate between your wireless clients and your wired LAN.

Where you will plug in the LAN interface will depend on your chosen network design. The next sections cover your options and your considerations in which to choose.

# Bridging wireless to your LAN

One common means of deploying wireless is to plug the access point directly into the same switch as your LAN hosts, where the AP bridges the wireless clients onto the wired network. This will work fine, but offers limited control over your wireless clients' ability to communicate with your internal systems.

# Bridging wireless to an OPT interface

If you want more control over your wireless clients, adding an OPT interface to pfSense for your access point is the preferred solution. If you wish to keep your wireless and wired networks on the same IP subnet and broadcast domain, you can bridge the OPT interface to your LAN interface. This scenario is functionally equivalent to plugging the access point directly into your LAN switch, except since pfSense is in the middle, it can filter traffic from your wireless network to provide protection to your LAN hosts.

You can also put your wireless network on a dedicated IP subnet if desired, by not bridging the OPT interface on pfSense and assigning it with an IP subnet outside of your LAN subnet. This enables

routing between your internal and wireless networks, as permitted by your 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 you will force your wireless clients to connect to a VPN before allowing connections to internal network resources.

## Choosing bridging or routing

The choice between bridging (using the same IP subnet as your LAN) or routing (using a dedicated IP subnet for wireless) for your wireless clients will depend on what services your wireless clients require. Certain applications and devices rely on broadcasts to function. Apple's AirTunes, as one example, will not function across two broadcast domains, so if you have AirTunes on your wireless network and want to use it from a system on your wired network, you must bridge your wired and wireless networks. Another example is media servers used by devices such as TiVo, Xbox 360, and Playstation 3. These rely on multicast or broadcast traffic that can only function if your wired and wireless networks are bridged. In many home network environments you will have applications or devices that require your wired and wireless networks to be bridged. In most corporate networks, there aren't any applications that require bridging. Which to choose depends on the requirements of network applications you use, as well as your 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. Having a WINS (Windows Internet Name Service) server is another example, as it will allow you to browse networks of Windows/SMB-enabled machines even when you are not in the same broadcast domain.

# pfSense as an Access Point

With a wireless card that supports hostap mode (See the section called “Cards Supporting Access Point (hostap) Mode”), pfSense can be configured as a wireless access point.

# Should I use an external AP or pfSense as my access point?

Historically, the access point functionality in FreeBSD has suffered from serious compatibility problems with some wireless clients. With FreeBSD 7.x it improved significantly, and again with FreeBSD 8.x, however there may still be some incompatible devices. FreeBSD 9.x offers even better support, but when development of pfSense 2.1 began, FreeBSD 9.x was not considered a stable enough base. These difficulties with client compatibility are not always limited to FreeBSD, but you may find that a cheap consumer grade wireless router turned access point provides better compatibility than FreeBSD's access point capabilities in some instances. We use pfSense access points at home with no trouble, with gear such as a MacBook Pro, Apple AirTunes, Mac mini G4, iPod Touch, iPad, Palm Treo, Android phones and tablets, various Windows laptops, Xbox 360, and FreeBSD clients and it works very reliably across all these devices. There is the possibility of finding incompatible devices with any access point. FreeBSD is no exception and you may find this is more common with FreeBSD than other access points. In older versions of FreeBSD, particularly with m0n0wall on FreeBSD 4.x, I recommended not using FreeBSD access point functionality. Today it works well with almost every device and is probably suitable for your network.

This is subject to significant change with each FreeBSD release. Our coming 2.2 release on FreeBSD 10 includes major wireless improvements that should make this concern a thing of the past. An up to date listing of known incompatible devices and the most recent information on wireless compatibility can be found at http://www.pfsense.org/apcompat.

As mentioned earlier, the main deciding factor for many people these days is 802.11n support. Because pfSense does not have full support for 802.11n, the speed of wireless is limited to 802.11g rates. This is a deal breaker for some, and as such using an external access point would be best for networks requiring 802.11n.

# 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 should be familiar if you have configured other wireless routers before, and some options may be new unless you have used some commercial-grade wireless equipment. There are dozens of ways to configure access points, and they all depend upon your environment. Here, we cover setting pfSense up 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 doing anything else, ensure that the wireless card is in the router, and the antenna is firmly attached. As described earlier in this chapter, 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 it "WLAN" (Wireless LAN) or "Wireless" will make it easy to identify in the list of interfaces. If you have a unique SSID, you may find it more convenient to use that in the description instead. If pfSense will be driving multiple access points, there should be some way to distinguish them, such as "WLANadmin" and "WLANsales". We'll call this one **`ConfRoom`** for now.

## Interface Type/IP Address

Since this will be an access point on a dedicated IP subnet, you will need to set the IPv4 Configuration Type to **`Static IPv4`** and specify an IPv4 Address and subnet mask. Since this is a separate subnet from the other interfaces, 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 example interface will be *`192.168.201.1`*.

## Common Wireless Settings

These settings are held in common for a physical wireless card, and are shared between any virtual wireless interfaces on the same card. Changing these settings on one interface will change them on all other interfaces using the same physical adapter.

### Persist common settings

By checking Persist common settings, the settings in this section will be preserved if the interfaces 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`**, and **`802.11a turbo`**, and possibly others. For this example, we will choose **`802.11g`**.

### 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. Unless you are having issues in such an environment, it is best left disabled. There is a performance penalty for using it, since it has some overhead on each frame and also requires some extra steps when transmitting frames.

### Transmit power

Only supported on certain cards, the Transmit power value will attempt to configure the card to transmit at the power specified. Some cards/drivers will round this value to the nearest supported

power level. This setting will not let you exceed the power level specified by the regulatory domain that limits the actual power used by the card.

## Wireless Channel Selection

When selecting a Channel, you will need to be aware of any nearby radio transmitters in similar frequency bands. 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 you can get away with using any channel you like, as long as your AP clients are near the antenna. The safest channels to use are **1**, **6**, and **11** since their frequency bands do not overlap each other. You may specify **Auto** to tell the card to pick an appropriate channel, however this functionality does not work with some wireless cards. If you choose **Auto** and things do not work, choose a specific channel instead. For this network, since there are no others around, we'll choose channel **1**. When using other standards, or using wireless in countries other than the US, you may find there are 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 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.

## Antenna settings

If supported by your wireless card, this setting may allow you to manually select which antenna is used to transmit and/or receive if needed.

## Distance setting

Measured in meters, and only supported by Atheros cards, This field can be used to tune ACK/CTS timers to fit the distance between AP and Client. Mostly it is not necessary to configure this, but it may help in certain tricky wireless setups.

## Regulatory settings

The Regulatory settings section controls how the card is allowed to transmit legally in your region. Different countries typically have different regulatory settings, and some countries have none. If you are unsure, you can check with your government to see which laws apply to you. The default values are usually OK, as the cards may be set to a specific region already. In some cases you must set them manually if the card has a default not understood by the driver.

While it may be tempting to set the card to **Debug** in order to use settings that are not otherwise allowed, you may find yourself in legal trouble should it get noticed. The likelihood of this happening varies greatly by country/area so use that with caution.

### Regulatory domain

This is the governmental body that controls wireless communications in your region. For example, the US and Canada follow FCC regulations while in the UK it's ETSI. If you are unsure of your regulatory domain, see the Country setting.

### Country

Sometimes specific countries inside a regulatory domain have even different restrictions. This 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 AP's 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 have any effect on 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)

This will be the "name" of the AP as seen by clients. You should set the SSID to something readily identifiable, yet unique to your setup. Keeping with the example, this can be named **`ConfRoom`**.

## Minimum wireless standard

The Minimum wireless standard setting 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 mobile devices such as the Nintendo DS and the Palm Tungsten C are only compatible with 802.11b and require a mixed network in order to work. The flip side of this is that you will see slower speeds as a result of allowing such devices on your network, as the access point will be forced to cater to the lowest common denominator when an 802.11b device is present. In our example conference room, people will only be using recently purchased company-owned laptops that are all capable of 802.11g, so we will check this option.

## Intra-BSS Communication

If you check Allow intra-BSS communication, wireless clients will be able to see each other directly, instead of routing all traffic through the AP. If clients will only need access to the Internet, it is typically safer to uncheck this. In our scenario, people 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 over wireless. This feature is not supported by all cards/drivers. In most cases, this should be left off, but can be enabled if you experience issues delivering audio or video over wireless.

## Enable Hide SSID (Disable SSID Broadcasting)

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 you have a wireless network available, but in most cases the convenience outweighs the 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

Three types of encryption are supported for 802.11 networks: WEP, WPA, and WPA2. WPA2 with AES is the most secure. Even if you are not worried about encrypting the over-the-air traffic (which you should be), it provides an additional means of access control. A WPA/WPA2 passphrase is also easier to work with then a WEP key on most devices; it acts more like a password than a really long string of hexadecimal characters. As with the choice between 802.11b and 802.11g, some older devices only support WEP or WPA, but most modern wireless cards and drivers will support WPA2.

For our conference room, they will use WPA2, and turn off WEP. To do this, uncheck Enable WEP, and check Enable WPA. To ensure that only WPA2 will be in use, set WPA Mode to **`WPA2`**. For our

WPA Pre-Shared Key, we'll use **excoconf213**, and also set WPA Key Mode Management to **Pre-Shared Key**.

To use WPA2+AES, as desired for the conference room wireless, set WPA Pairwise to **AES**.

### Note

To use WPA2 on a Windows XP wireless client, you must have a wireless driver that supports WPA2. If you are using Windows XP's Wireless Configuration interface, in order to associate with an access point running WPA2 you will need to upgrade the PC to Windows XP SP3 or install the patch from Microsoft Knowledge Base article 917021 [http://support.microsoft.com/kb/917021].

#### Wireless encryption weaknesses

WEP has had serious known security problems for years now, and should never be used unless it is the only option for wireless devices you must support. It's possible to crack WEP in a matter of minutes at most, and it should never be relied upon for security. WEP cannot be relied upon for anything more than keeping out Internet seekers with no technical skills.

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. Recently these attacks are becoming more practical. At the time of this writing it isn't nearly as easy to break as WEP, but you should still never use it unless you have devices that are incompatible with WPA or WPA2 using AES. WPA and WPA2 in combination with AES are not subject to these flaws in TKIP.

### Key Rotation

The Key Rotation option allows you to set 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 longer than the Master Key Regeneration value. The default value of *60* seconds (one minute) is adequate.

### Master Key Regeneration

The Master Key Regeneration parameter controls how often, in seconds, the master key used internally (Group Master Key, GMK) to generate GTKs is regenerated. It can be any value from *1* to *9999* but it should be higher than the Key Rotation value. The default value of *3600* seconds (one hour) is adequate.

### Strict Key Regeneration

The Strict Key Regeneration option makes the firewall change the GTK whenever a client leaves the access point. Much like changing the passwords when an employee leaves, it is a good idea to do this in most cases. There may be a slight performance penalty in cases where you have 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 wireless security that is supported 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 you already have an existing RADIUS server, or can easily configure one, it may be a viable source of wireless access control. In this example, we will not set or use 802.1X, but the options are explained.

### 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.

#### 802.1X Authentication Server IP Address

This option defines the preferred RADIUS server to use for authentication of 802.1X clients.

#### 802.1X Authentication Server Port

The port upon which to contact the RADIUS server for authentication requests (Typically *1812*)

#### 802.1X Authentication Server Shared Secret

The password to use when communicating with the RADIUS server from this firewall.

#### Secondary 802.1X Authentication Settings

These define the same parameters as above, but for a secondary RADIUS server in case the first one is unreachable.

#### 802.1X 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 should be enough to get a wireless access point running with 802.11g with WPA2 + AES encryption. There are other settings that can be used to fine-tune the AP's behavior, but they are not necessary for normal operation in most environments. When you have finished changing the settings, click Save, then Apply Changes.

## Configuring DHCP

Now that we have created an entirely separate network, we will want to enable DHCP so that associating wireless clients can automatically obtain an IP address. Browse to Services → DHCP Server, click on the tab for your wireless interface (ConfRoom for our example setup). Check the box to enable, set whatever size range you will need, and any additional needed options, then click Save and Apply Changes. For more details on configuring the DHCP service, see the section called “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 you will need to have a rule to allow traffic from this subnet to whatever destination will be needed. Since our 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, go to Firewall → Rules, and click on the tab for the wireless interface (ConfRoom for this example). Add a rule to pass traffic of any protocol, with a source address of the ConfRoom subnet, and any destination. For more information about creating firewall rules, see Chapter 10, *Firewall*.

## Associating Clients

The newly configured pfSense AP should appear in the list of available access points from your wireless device, assuming you did not disable broadcasting of the SSID. You should be able to associate clients with it as you would 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 you have a wireless interface configured for access point mode, the associated clients will be listed on Status → Wireless.

# Additional protection for your wireless network

In addition to strong encryption from WPA or 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 VPN. These methods can be employed whether you use an external access point on an OPT interface or an internal wireless card as your access point.

## Additional wireless protection with Captive Portal

By enabling Captive Portal on the interface where your wireless resides, you can require authentication before users can access network resources. 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 Chapter 24, *Captive Portal*.

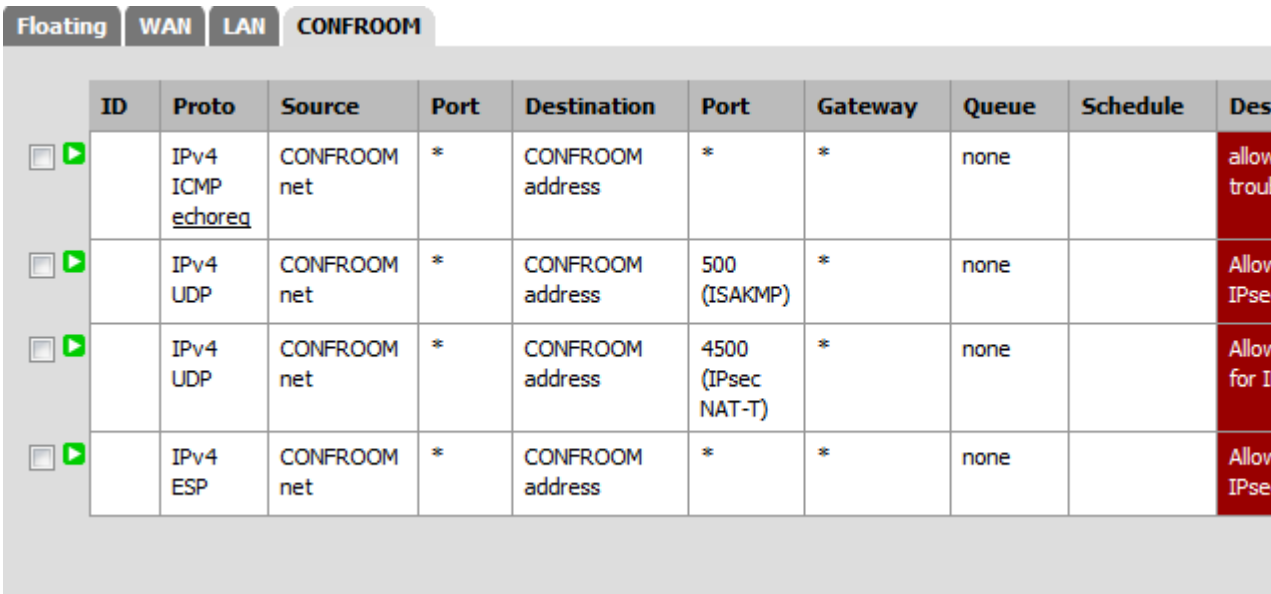
## Additional protection with VPN

Adding Captive Portal provides another layer of authentication, but does not offer any additional protection from eavesdropping of your wireless traffic. Requiring VPN before allowing access to the internal network and Internet adds another layer of authentication as well as an additional layer of encryption for your wireless traffic. The configuration for your chosen type of VPN will be no different from a remote access configuration, but you will need to configure the firewall rules on the pfSense interface to only allow VPN traffic from your wireless clients.

### Configuring firewall rules for IPsec

Figure 23.6, “Rules to allow only IPsec from wireless” shows the minimal rules required to allow only access to IPsec on the WLAN interface IP. Pings to the WLAN interface IP are also allowed to assist in troubleshooting.

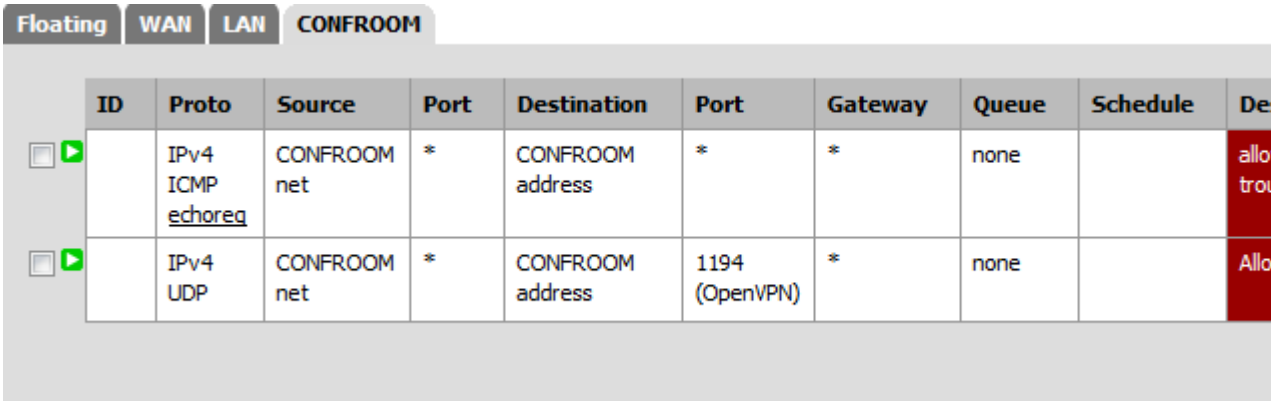
**Figure 23.6. Rules to allow only IPsec from wireless**



## Configuring firewall rules for OpenVPN

Figure 23.7, “Rules to allow only OpenVPN from wireless” shows the minimal rules required to allow access only to OpenVPN on the WLAN interface IP. Pings to the WLAN interface IP are also allowed to assist in troubleshooting. This assumes you are using the default UDP port 1194. If you choose another protocol or port, adjust the rule accordingly.

**Figure 23.7. Rules to allow only OpenVPN from wireless**




# Configuring a Secure Wireless Hotspot

Your company or organization may wish to provide Internet access for customers or guests using your existing Internet connection. This can be a boon to your customers and business, but can also expose your private network to attack if not done properly. This section covers the common means of providing Internet access to guests and customers, while protecting your internal network.

## Multiple firewall approach

For the best protection between your private network and public network, obtain at least two public IPs from your ISP, and use a second firewall for your public network. To accommodate this, you put a switch between your Internet connection and the WAN of both firewalls. This also has the benefit of putting your public network on a different public IP from your private network, so if you should receive a report of abuse, you will be able to easily differentiate whether it originated from your public or private network. The firewall protecting your private network will see your public network no differently than any Internet host.

## Single firewall approach

In environments where the multiple firewall approach is cost prohibitive or otherwise undesirable, you can still protect your internal network by connecting your public network to an OPT interface on pfSense. You should assign a dedicated private IP subnet to this OPT interface, and configure your firewall rules to allow access to the Internet but not your internal network.

## Access control and egress filtering considerations

Other than not allowing traffic from the publicly accessible network to the private network, there are additional things you should consider in the configuration of your hotspot.

### Restrict network access

While many hotspots use open wireless networks with no other authentication, you should consider additional protections to prevent network abuse. On your wireless, consider using WPA or WPA2 and providing the passphrase to your guests or customers. Some will have 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 Chapter 24, *Captive Portal*). This helps prevent people in other offices and outside the building from using your wireless network.

### Disable Intra-BSS communication

If your access point allows, you should not allow intra-BSS communication. This prevents wireless clients from communicating with other wireless clients, which protects your users from intentional attacks from other wireless users as well as unintentional ones such as worms.

### 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 you should consider additional restrictions. To avoid having your public IP address black listed because of infected visiting systems acting as spam bots, you should 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 email 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 on the Firewall: Rules: Edit page. 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, you won't want to set this too low to prevent blocking legitimate users, but a limit of five connections should be reasonable. You may wish to specify Maximum state entries per host on all your 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 your limits too low.

You will need to balance the desires of your users against the risks inherent in providing Internet access for systems you do not control, and define a policy that fits your 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 them a few times, you may need to replace them.

## Try with multiple clients or wireless cards

To eliminate a possible incompatibility between pfSense's wireless functions and your 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 you have a weak signal, even when you are nearby the access point antenna, check the antenna again. For mini-PCI cards, if you only have one pigtail in use and there are two internal connectors,

try hooking up to the other internal connector on the card. You can also try changing the Channel or adjusting the Transmit Power, or the Antenna Settings on the wireless interface configuration. For mini-PCI cards, check for broken ends on the fragile pigtail connectors where they plug into the mini-PCI card.

# Chapter 24. Captive Portal

The Captive Portal feature of pfSense allows you to direct users to a web page before Internet access is permitted. From that page, you can either let users access the Internet after clicking through, or require authentication. The most common uses of Captive Portal are for wireless hot spots, or 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

pfSense 2.0 removed most of the limitations in the Captive Portal from 1.2.3, and added support for vouchers, which are pre-generated access codes that can be provided to users to grant them short-term access.

pfSense 2.1 introduced multiple Captive Portal zones, and brought significant enhancements in efficiency.

# 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. We are planning to have complete IPv6 support for Captive Portal in pfSense 2.2.

## Not capable of reverse portal

A reverse portal, requiring authentication for traffic coming into your network from the Internet, is not possible.

# Captive Portal Zones

New in pfSense 2.1 are zones for the Captive Portal. This feature allows you to have separate portals for different sets of interfaces. For examples, LAN and WLAN could get one portal, while a Conference Room could get a separate portal page. Each zone can have separate settings for HTML pages, authentication, allowed addresses, etc. This leads to an extra step in the portal management process, in that you need to first create a zone before you can setup a portal.

For users upgrading from an earlier version of pfSense, a zone will be created for you automatically.

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

When you first visit Services → Captive Portal, you are presented with a list of Captive Portal zones, if any exist. On the very first visit to the page, the list will be empty. You may create a new zone by clicking .

To create a zone you must enter a Zone Name and optionally a Description. The Zone Name may only consist of upper or lowercase letters, numbers or an underscore; You cannot use spaces or other special characters. The description may be formatted as desired. When you click Continue, you are forwarded on to the Captive Portal configuration screen for this new zone.

From the zone list, you can edit the settings for an existing zone by clicking . A zone can be completely removed by clicking and then OK on the confirmation dialog.

# Common Captive Portal Scenarios

The following are some common basic scenarios for using 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, all you need to do is create a new zone, check the Enable captive portal box, select an interface, and upload a HTML page with your portal contents as described in the section called “Portal page contents”. You may wish to specify additional configuration options as detailed in the section called “Zone Configuration Options”.

## Portal Configuration Using Local Authentication or Vouchers

To setup a portal with local authentication, create a zone, check the Enable captive portal box, select an interface, choose Local User Manager / Vouchers, and upload a HTML page with your portal contents as described in the section called “Portal page contents”. You may wish to specify additional configuration options as detailed in the section called “Zone Configuration Options”. Then configure your local users in the User Manager Chapter 7, *User Management and Authentication*.

If you wish to use Vouchers, then proceed to the vouchers tab and create them there. See the section called “Vouchers” for more information on Vouchers.

## Portal Configuration Using RADIUS Authentication

To setup a portal using RADIUS authentication, first configure your RADIUS server, then follow the same procedures as setting up a portal with local authentication, filling in the appropriate information for your RADIUS server, and select RADIUS authentication. Read the next section for information on specific configuration options you may wish to use.

# Zone Configuration Options

This section describes each of the Captive Portal configuration options. These options area available once you have created your first captive portal zone under Services → Captive Portal. These options all work independently of one another for each zone. For example, the allowed IP addresses in one zone are only allowed in that single zone, and no other, unless you add them there also.

## Interface

Here you select the interfaces that will be active for this Captive Portal zone. This cannot be any WAN or OPT 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

This field specifies the maximum number of concurrent connections per IP address. The default value is 4, which should suffice for most environments. This limit exists to prevent a single host from exhausting all resources on your 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 your system unresponsive.

# Idle timeout

If you want to disconnect idle users, fill in a value here. Users will be able to log back in immediately.

# Hard timeout

To forcefully log off users after a specified period, enter a hard timeout value. You should enter either a hard timeout, idle timeout or both to ensure sessions are removed if users do not log off, as most 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 you set timeout values, make sure that the timeout is less than the DHCP lease time, or you can end up with captive portal sessions that are 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

Pass-through credits let you give a grace period to devices before they must authenticate via the portal. For example, a device could connect 3 times within a day without even 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, you should set a hard timeout and/or idle timeout.

## Pass-through credits allowed per MAC address

This setting defines how many times per MAC address a connection will be passed 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

Clients will have their available pass-through credits restored to the original count after this amount of time, specified in hours, since 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

Check this box to enable a logout pop up window 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 of your users unless you control the computers and can exclude your portal in their pop up blocker.

# Pre-authentication redirect URL

The pre-authentication redirect URL, as the name implies, redirects users to the URL you specify before they authenticate. Commonly, this is used to have a custom landing page describing your location hosted on a server locally or elsewhere (See the section called “Allowed Hostnames” to allow hostnames through the portal without authentication, and the section called “Allowed IP Address” for IP addresses). That landing page would have a link which in turn redirects the users back to your portal page, e.g. *`http://x.x.x.x:8000/index.php`*.

Your custom captive portal page will need to have some 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["REQ
$portal_redirurl = urldecode("$PORTAL_REDIRURL$");
if(!stristr(urldecode("$PORTAL_REDIRURL$"), $request_uri)) {
 Header("Location: $PORTAL_REDIRURL$");
 exit;
}
?>
```

# After authentication Redirection URL

If you enter a URL here, 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 tried to access.

# Concurrent user logins

If this box is 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

This option allows you to disable the default MAC filtering. This is necessary in cases where the MAC address cannot reliably be determined, such as when multiple subnets exist behind a separate router using the portal, as all users behind a router will show up to the portal as 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. If this is enabled, RADIUS MAC authentication cannot be used.

# Pass-through MAC Auto Entry

Some people only want users to authenticate once per device, and then never see the portal login again unless they change devices. Setting up pass-through MAC entries automatically can 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 you either have to log in and remove it manually from the Pass-through MAC tab or send a POST from another system to remove it. If this is enabled, RADIUS MAC authentication cannot be used. Also, 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 you either have to log in and remove it manually from the Pass-through MAC tab or send a POST from another system to remove it.

# 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 let you define the default values for user bandwidth, specified in Kilobits per second. These values can be overridden by RADIUS (the section called “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 you to configure authentication if desired. If you require authentication, you can use either the local user manager or RADIUS authentication.

## No Authentication

If you leave No authentication selected, users will just have to click through your portal screen for access. The form must still be submitted, but you do not need to have any user entry fields, only a submit button.

## Local User Manager / Vouchers

If you wish to have users authenticate with a username and password, but do not want to use a RADIUS server, you can manage your Captive Portal users in the pfSense GUI by selecting Local User Manager / Vouchers as the authentication type. Local users are added in the User Manager Chapter 7, *User Management and Authentication*.

There is also an optional Captive Portal user permission that can be used to restrict portal access to only a certain set of users. If you check Allow only users/groups with 'Captive portal login' privilege set, then the 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 them short-term access. Vouchers may be used in addition to, or instead of, local user authentication. For more information on using vouchers, see the section called “Vouchers” later in this chapter.

## RADIUS Authentication

RADIUS is a means of authentication against a central server that contains account information. There are many implementations of RADIUS out there, such as FreeRADIUS, Radiator, and IAS/NPS on Windows servers. For those with a Microsoft Active Directory network infrastructure, RADIUS can be used to authenticate captive portal users from your Active Directory using Microsoft IAS or NPS. This is described in the section called “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 your RADIUS server for more information.

To use RADIUS Authentication, select that as your Authentication option and then fill in the data about your RADIUS server.

One of the first things you need to know is which RADIUS Protocol that your RADIUS server can use for authentication. Your 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, but has since been broken completely, so it is no longer considered trustworthy. See the section called “PPTP Security Warning”. |

The relative (in)security of these protocols may be of little consequence depending on the layout of your network and the location of the RADIUS server, but should still be considered.

## Passing back configuration from RADIUS Servers

Some of the 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. |
| Session Terminate Time | Works similarly to *`Session-Timeout`*, but drawn from the *`WISPr-Session-Terminate-Time`* attribute. |
| 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`*. |
| Reply Message | *`Reply-Message`* from RADIUS will add a the text of that attribute as a note to the portal auth log for the user. |

## 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 this:

```
<tr><td align="right">Username:</td><td><input name="auth_user" type="text" sty
<tr><td align="right">Password:</td><td><input name="auth_pass" type="password"
```

This is what most people will use. 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 client's shared secret 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, you can define a primary server and secondary server to be used if the primary fails.

The secondary authentication source uses the form fields *`auth_user2`* and *`auth_pass2`* in the captive portal HTML, such as this

```
<tr><td align="right">Username:</td><td><input name="auth_user2" type="text" sty
<tr><td align="right">Password:</td><td><input name="auth_pass2" type="password
```

## Accounting

RADIUS accounting will send information back to the RADIUS server about when a user's session starts, ends, and how much data they have transmitted. Not all RADIUS servers support or are configured to accept accounting data, so make sure that you have set your RADIUS server 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 your server. | |
| | No accounting updates | Synonymous with disabling accounting, it will not send accounting updates to the server. |
| | Stop/start accounting | Will send START and STOP records for a user's session only. |
| | Interim update | Will send START and STOP records and also periodically send updates to the server while a user's session is active. This is less likely to lose session data should the firewall restart 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 reauthentication is enabled, *`Access-Requests`* 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 you to actively terminate user sessions from the RADIUS server.

**Note**

If you use concurrent login limits 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.

RADIUS MAC authentication

If this option is enabled, the captive portal will try 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 you have Disable MAC filtering checked.

RADIUS NAS IP attribute

This field controls what is sent to the RADIUS server in the *Calling-Station* attribute. Choose the firewall's Interface/IP address you wish 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**, in *Access-Requests* the value of *Calling-Station-Id* will be set to the client's IP address and the *Called-Station-Id* to the client's MAC address. **Default** behavior is *Calling-Station-Id* = client's MAC address and *Called-Station-Id* = pfSense's 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. Here you can specify a NAS Identifier to override the default value

MAC address format

This option changes the MAC address format used in RADIUS. Change this if you also need to change 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*. |
| singledash | 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 you check this you must also choose an SSL Certificate.

## HTTPS server name

This field is where you specify the FQDN (hostname + domain) to be used for HTTPS. This needs to match the Common Name (CN) on your certificate to prevent your users from receiving certificate errors in their browsers.

## SSL Certificate

Here you select your SSL certificate to be used by the portal for HTTPS logins. Certificates are managed in Chapter 8, *Certificate Management*.

# Portal page contents

Here you upload a HTML page containing the portal page your users will see when trying to access the Internet before authenticating or clicking through the portal.

## Portal page without authentication

This shows the HTML of a portal page that can be used without authentication.

```
<html>
<head>
<title>Welcome to our portal</title>
</head>
<body>
<p>Welcome to our portal</p>
<p>Click Continue to access the Internet</p>
<form method="post" action="$PORTAL_ACTION$">
  <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
  <input name="zone" type="hidden" value="$PORTAL_ZONE$">
  <input name="accept" type="submit" value="Continue">
</form>
</body>
</html>
```

## Portal page with authentication

Here is an example portal page requiring authentication.

```
<html>
<head>
<title>Welcome to our portal</title>
</head>
<body>
<p>Welcome to our portal</p>
<p>Enter your username and password and click Login to access the Internet</p>
<form method="post" action="$PORTAL_ACTION$">
  <input name="auth_user" type="text">
  <input name="auth_pass" type="password">
  <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
  <input name="zone" type="hidden" value="$PORTAL_ZONE$">
  <input name="accept" type="submit" value="Login">
</form>
```

```
</body>
</html>
```

## Portal page with Vouchers

Here is an example portal page for use with vouchers.

```
<html>
<head>
<title>Welcome to our portal</title>
</head>
<body>
<p>Welcome to our portal</p>
<p>Enter your voucher code and click Login to access the Internet</p>
<form method="post" action="$PORTAL_ACTION$">
  <input name="auth_voucher" type="text">
  <input name="redirurl" type="hidden" value="$PORTAL_REDIRURL$">
  <input name="zone" type="hidden" value="$PORTAL_ZONE$">
  <input name="accept" type="submit" value="Login">
</form>
</body>
</html>
```

# Authentication error page contents

Here you can upload a HTML page to be displayed on authentication errors. 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:

```
<HTML>
<HEAD><TITLE>Redirecting...</TITLE></HEAD>
<BODY>
<SPAN STYLE="font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif; font-s
<B>Redirecting to <A HREF="<?=\$my_redirurl;?>"><?=\$my_redirurl;?></A>...</B>
</SPAN>
<SCRIPT LANGUAGE="JavaScript">
<!--
LogoutWin = window.open('', 'Logout', 'toolbar=0,scrollbars=0,location=0,statusb
if (LogoutWin) {
 LogoutWin.document.write('<HTML>');
 LogoutWin.document.write('<HEAD><TITLE>Logout</TITLE></HEAD>') ;
 LogoutWin.document.write('<BODY BGCOLOR="#435370">');
 LogoutWin.document.write('<DIV ALIGN="center" STYLE="color: #ffffff; font-famil
 LogoutWin.document.write('<B>Click the button below to disconnect</B><P>');
 LogoutWin.document.write('<FORM METHOD="POST" ACTION="<?=\$logouturl;?>">');
 LogoutWin.document.write('<INPUT NAME="logout_id" TYPE="hidden" VALUE="<?=\$ses
 LogoutWin.document.write('<INPUT NAME="zone" TYPE="hidden" VALUE="<?=\$cpzone;?
 LogoutWin.document.write('<INPUT NAME="logout" TYPE="submit" VALUE="Logout">')
 LogoutWin.document.write('</FORM>');
 LogoutWin.document.write('</DIV></BODY>');
 LogoutWin.document.write('</HTML>');
```

```
 LogoutWin.document.close();
}

document.location.href="<?=\$my_redirurl;?>";
-->
</SCRIPT>
</BODY>
</HTML>
```

Because browsers have pop-up blockers that will most likely stop that logout window from appearing, you may want to investigate other possible means of creating a JavaScript pop-up using similar code.

# Pass-Through MAC

The Pass-Through MAC tab lets you specify MAC addresses that should be passed straight through the portal for this zone without requiring authentication.

| | |
|---|---|
| MAC address | The MAC address of the device to allow. The value should 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. Leave blank to not specify a limit. |

# Allowed IP Address

The Allowed IP Address tab works similarly to Pass-Through MAC tab, except it works by IP address instead of MAC address. The device with the specified IP address 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. |
| 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 you enter them by hostname instead of IP. A process will continually resolve the IP address(es) of the hostname and allow them through the portal without authentication in this zone. The most common use of this is to make a "walled garden" style portal, where the users can still access a few sites without authenticating. This is also commonly used with the Pre-authentication Redirect URL if it 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, you may have to add quite a few more hosts than you expect to this list.

| | |
|---|---|
| Direction | Use **`From`** to always allow a given Hostname through the captive portal (without authentication). Use **`To`** to allow access from all clients (even non-authenticated ones) behind the portal to this Hostname. **`Both`** will allow traffic in both directions. Depending on how you intend to use this feature, you most likely will want the **`To`** or **`Both`** option. |
| Hostname | The fully qualified domain name (FQDN) of the target host or site. |

| | |
|---|---|
| Description | Some text describing the entry, if desired. |
| Bandwidth up/down | The amount of bandwidth that this device may use. Leave blank to not specify a limit. |

# Vouchers

Vouchers are special codes that can be used to gain Internet access through the Captive Portal. Each roll of vouchers is cryptographically generated and includes a set time limit. Vouchers are commonly implemented in places authenticated, but time-limited, Internet access is desired without needing to provide a username and password to users. For example, in coffee shops, hotels, and airports. Users simply enter their voucher code in the portal login form and are granted access for as long as the voucher is valid. 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 are only good from the start of the session, for the duration of the voucher length. 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.

The voucher settings are unique per Captive Portal Zone, so you can get to their settings under Services → Captive Portal, edit your zone, and then go to the Vouchers tab. The options on the page are initially disabled, you must click Enable Vouchers to make changes on the remainder of this page.

Before vouchers can be used, a public/private RSA key pair must be generated. A set is generated for you the first time the page is visited that is 32-bits in length, but you may generate your own pair manually if you wish. The maximum key length supported is 64 Bits. Using shorter keys will make the generated vouchers shorter but eventually less secure.

In most cases, the remaining options on this screen can simply 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 create new rolls. No options appear here until after the other settings have been configured once and saved. See the section called “Managing Voucher Rolls” for more on managing voucher rolls. |
| Voucher Keys | Before you have enabled vouchers, take note that the Voucher Public Key and Voucher Private Key are not standard defaults we included but are randomly generated each time the page loads while vouchers are disabled. Once you enable vouchers and save, the keys are set. |

## Note

Take care not to change the keys or other bits once you have set them, or you will render all current vouchers invalid and you will need to create new vouchers.

| | |
|---|---|
| Voucher Public Key | This key is used to decrypt vouchers. You can use the pre-generated key, click Generate to make a new public and private key, or make a key elsewhere and paste an RSA public key (64 Bit or smaller) in PEM format here. |

Voucher Private Key — This key is only used to generate encrypted vouchers and doesn't need to be available if the vouchers have been generated offline. You can use the pre-generated key, click Generate to make a new public and private key, or make a key elsewhere and paste an RSA private key (64 Bit or smaller) in PEM format here.

Character Set — The character set lets you define what characters are valid for voucher text. It is case sensitive and should contain printable characters (numbers, lower case and upper case letters) that are hard to confuse with others. For example, you should avoid 0 (Digit zero), O (Letter O), l (Lowercase L), and 1 (Digit One). It cannot contain a space, double quote, or comma.

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 to suit your purpose if you wish. The total of all these 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 if you need 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 when 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 when they enter a voucher that was valid, but has since expired.

Once you are satisfied with the settings on the page, click Save, and the page will reload with the voucher management options activated.

# Managing Voucher Rolls

Vouchers are created in batches called Rolls. Each roll has specific settings that are unique to that roll. For example, you can have a roll with an 8-hour time limit, and a roll with a 12-hour time limit, and hand out voucher codes depending on which level of service a person should receive.

## Creating Voucher Rolls

To create a roll, click just under the roll list. You will be presented with a screen with a few options to set for your new roll.

Roll #
: The number of this roll. Each roll should have a unique number. This can be any number from 0 to 65535.

Minutes per Ticket
: Defines how long the voucher lasts, in minutes. Remember, the clock starts the moment the voucher is used, and does not stop, so plan the voucher length accordingly. Because this is defined in minutes, ensure you have the correct length defined, e.g. 1440 minutes is 24 hours.

### Note

If you want the users to be disconnected when these vouchers expire, you must check the portal option Session Timeout or they will remain online indefinitely.

Count
: Defines how many vouchers will be made on this roll. It can be from 0 to 1023.

### Note

If you change the count on an existing roll, 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 your reference, such as *`2 hour vouchers for coffee purchases`*.

Click Save and your new roll will now be ready to use.

## Editing Existing Rolls

You can edit existing voucher rolls by clicking , but be careful — changing the Count while the roll exists will cause the old vouchers on the roll to be invalidated. Click Save when you are finished making changes.

## Removing Voucher Rolls

Rolls of vouchers can be removed by clicking at the end of their row. However, you probably want to only remove them if they have all been used, or if they should be deactivated for some reason (e.g. an employee snuck off with a roll's worth of printed codes).

If you remove a roll, *all* of the vouchers in that roll become invalid.

## Exporting/Downloading Voucher Rolls

To see the voucher codes, you have to download a voucher roll by clicking . This downloads a `.csv` (Comma Separated Value) spreadsheet containing the codes for this roll. You can save and then open this file up in the spreadsheet editor of your choice (LibreOffice Calc, Google Docs, Excel), and then do whatever you like with them. Print them off, feed the CSV file into a POS system, etc.

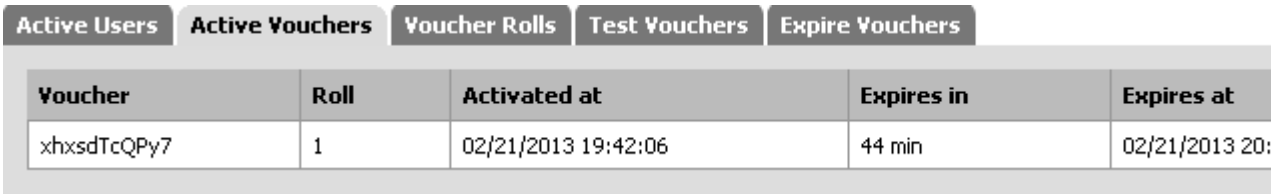
## Using Vouchers on Your Portal Page

Vouchers must be submitted via the `auth_voucher` form field. See the section called “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 24.1, “Active Vouchers”.

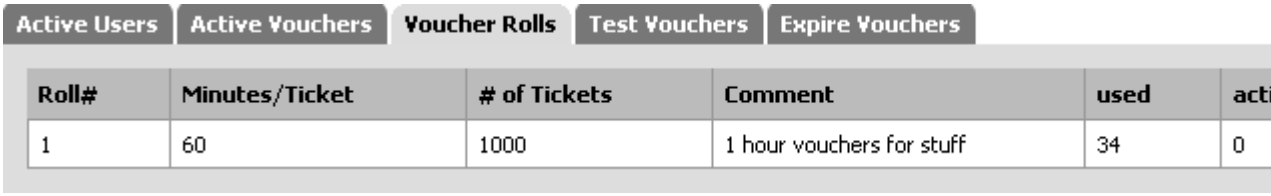
### Figure 24.1. Active Vouchers




## 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 24.2, “Vouchers Roll Usage”

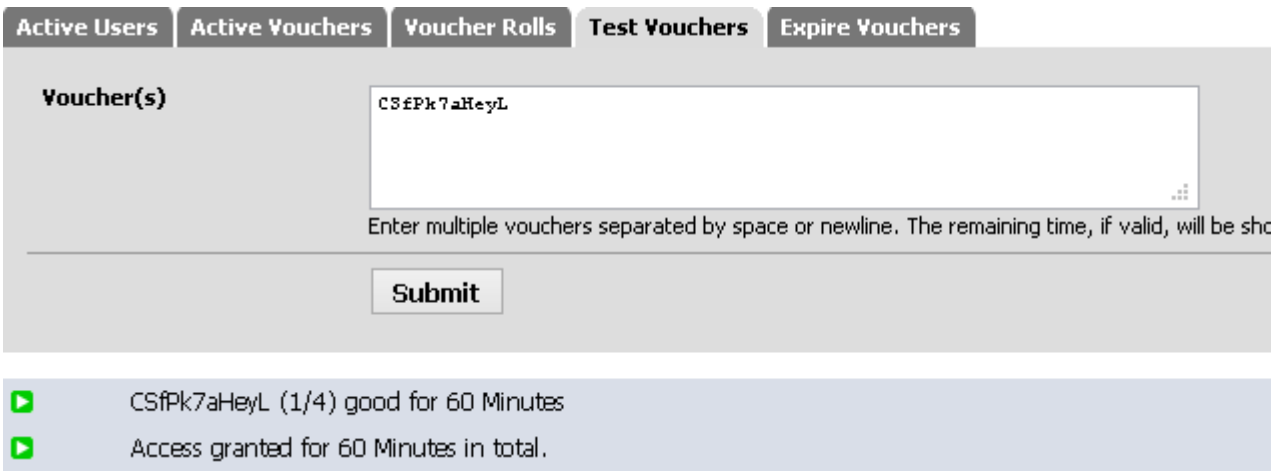
### Figure 24.2. Vouchers Roll Usage




## Testing Vouchers

You can test a voucher code to see if it is valid 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 how long that the voucher will work for, as seen in Figure 24.3, “Testing Vouchers”. Testing a voucher does not count it as used or expired, it is still free to be used at a later time.

### Figure 24.3. Testing Vouchers



## Expiring Vouchers

Vouchers can be invalidated while in use or before they have been used, in batches as large as you like, by entering them at Status → Captive Portal, on the Test Vouchers tab for a zone. After submitting, any voucher listed in the form will no longer work. Active vouchers are also immediately expired.

# 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 the section called “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 (Typically *admin*). |
| Voucher sync password | The GUI password for the target system. |

Unlike the configuration synchronization in a high availability cluster, you will need to configure this synchronization 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

You can manually create your own RSA public and private keys to use for vouchers on a separate system if desired. 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

If you need to have additional files for your captive portal page, such as style sheets, image files, PHP or JavaScript files, you may upload them in the File Manager tab under Services → Captive Portal and inside a zone.

The total size limit for all files in a zone is 1MB.

# File Name Conventions

When you upload a file here, the file's name will automatically be prefixed with `captiveportal-`. For example if you upload `logo.png`, 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. You may reference them directly from your portal page HTML code using relative paths. Example: you've uploaded an image with the name `captiveportal-logo.jpg` using the file manager. Then you can include it in your portal page like this:

```
<img src="captiveportal-logo.jpg">
```

You may upload PHP scripts as well, but you may need to pass them some extra parameters so they work as desired, for example:

```
<a href="/captiveportal-aup.php?zone=$PORTAL_ZONE$&redirurl=$PORTAL_REDIRURL$">A
```

## Uploading Files

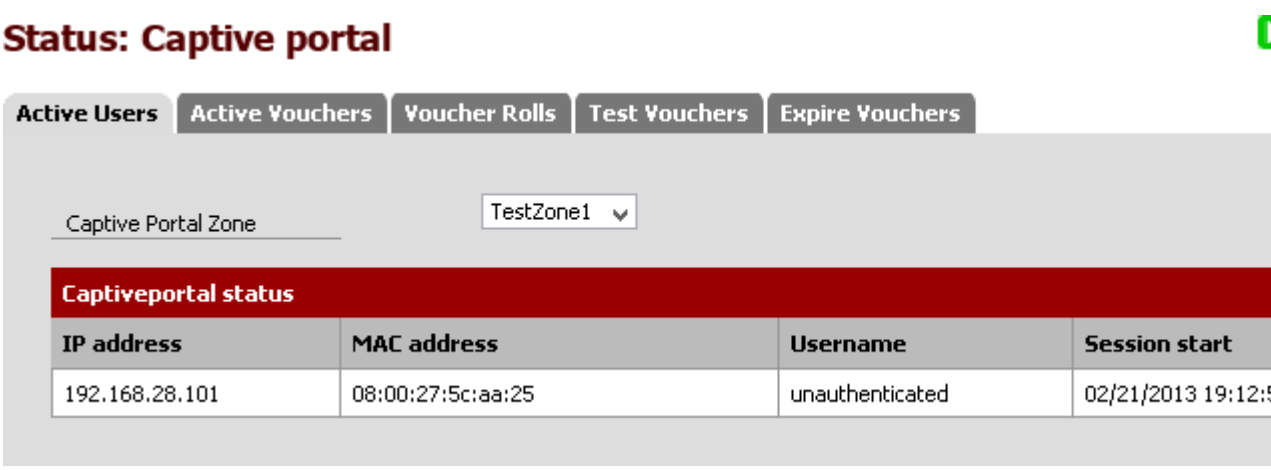
To upload files, click , then click Browse to find the file, and then Upload. The file will be transferred to the firewall and stored in the configuration.

# Viewing Authenticated Captive Portal Users

A list of currently active users for the portal can be found under Status → Captive Portal. On that page, you pick a zone first and then you will see a list of users online inside that zone. Depending on your authentication settings, you may see one of several different user styles.

If you do not use any authentication, you will see a listing like Figure 24.4, “Online Captive Portal Users — No Authentication”.

### Figure 24.4. Online Captive Portal Users — No Authentication




If you use username and password authentication (Local or RADIUS), you will see a listing like Figure 24.5, “Online Captive Portal Users — User Authentication”. If you use RADIUS with MAC Authentication, the username will be the MAC address.

### Figure 24.5. Online Captive Portal Users — User Authentication

| Captiveportal status | | | |
|---|---|---|---|
| IP address | MAC address | Username | Session start |
| 192.168.28.101 | 08:00:27:5c:aa:25 | jim | 02/21/2013 19:44:08 |

If you use vouchers, the list will look like Figure 24.6, “Online Captive Portal Users — Vouchers”.

### Figure 24.6. Online Captive Portal Users — Vouchers

| Captiveportal status | | | |
|---|---|---|---|
| IP address | MAC address | Username | Session start |
| 192.168.28.101 | 08:00:27:5c:aa:25 | xhxsdTcQPy7 | 02/21/2013 19:42:06 |

# 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 your RADIUS server, or problems on the RADIUS server itself. Check your RADIUS server's logs for indications of why access was denied, and ensure the firewall can communicate with the RADIUS server.

# Portal Page never loads (times out) nor will any other page load

This is most commonly a DNS failure. If you do not use the firewall for DNS, and you do not setup Allowed IP Address (the section called “Allowed IP Address”) entries 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 your firewall's IP and the DNS forwarder for client DNS, or add Allowed IP Address entries for your external DNS servers.

Another possible explanation for this is that your firewall rules on the interface with the portal do not allow the users to reach web sites on port 80. Ensure your firewall rules pass out traffic to TCP port 80, and to make sure DNS works, they must also be able to reach your DNS servers on TCP and UDP port 53.

# Users who load an HTTPS site as their home page do not get redirected to the captive portal login

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 your 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 to the portal

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 you may need to bring up a browser and manually navigate to an HTTP site to get the login redirect.

There have also been reports that on older version of OSX, 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 OSX.

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 into the portal

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 you want a port forward to always

work on a device behind the portal, then it must be setup to bypass the portal with either a Pass-through MAC entry (the section called “Pass-Through MAC”) or an Allowed IP Address entry (the section called “Allowed IP Address”).

# Voucher users online after their vouchers have expired

If you find that your voucher users are online after their vouchers have expired, the most likely cause is that the captive portal zone for the vouchers does not have Session Timeout enabled. When that setting is not enabled, the pruning process that clears expired sessions does not run, so users are left online indefinitely.

# Chapter 25. Firewall Redundancy / High Availability

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. This is provided by the combination of CARP, pfsync, and pfSense's XML-RPC configuration synchronization, each of which will be explained in this chapter. Often this is simply referred to as CARP, though technically CARP is only part of the complete solution. We'll refer to it as High Availability or HA.

Starting with pfSense 2.1, the High Availability synchronization Settings have been moved to System → High Avail. Sync. This was done to decouple the logical association of the synchronization settings from the Virtual IP settings because while they are somewhat related, they are independent functions. CARP VIPs can be used without doing failover, and synchronization can be used for other purposes besides CARP. Referring to the synchronization settings as "CARP" settings was a bit of a misnomer that led people to make false assumptions about the behavior and interaction between these functions.

It's important to distinguish between the three functions because they happen in different places. The config sync and pfsync traffic happens on your sync interface, directly communicating between the two firewall units. CARP heartbeats are sent on every interface with a CARP VIP, once per second by default, or varying depending on the advertisement skew and base. If a secondary unit fails to see a heartbeat from the master on any interface, it will attempt to take over as master. In other words, the failover signaling does not happen on the sync interface as one might believe, but rather it happens on every CARP-enabled interface.

## CARP Overview

Common Address Redundancy Protocol (CARP) was created by OpenBSD developers as a free, open redundancy solution for sharing IP addresses amongst 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.

Each pfSense firewall in a CARP group has its own unique IP address assigned on each interface, and has the shared CARP VIPs assigned as well. These CARP IPs are only active if the firewall is currently the master.

CARP sends out its own packets for other nodes to see, one for every VIP on every interface, once per second by default, depending on the values of the advertisement base and skew (more on that later). These heartbeats are sent via multicast and the switch will carry those on to the other members. CARP also monitors every interface that has a CARP VIP enabled watching for these CARP packets. Should a packet fail to arrive as fast as expected, it assumes a failure has taken place, and a lower priority node will take over as master. We configure the CARP system such that if a failure of any network interface is detected, the next designated firewall switches to master on all interfaces.

Either a loss of signal from the master on an interface (no heartbeats arrive) or if they arrive too slowly (slower than the secondary node's own rate), a failure will be assumed. This property of CARP's communication can be the cause of some issue at layer 2, because there are switches that either block multicast, handle it poorly, or otherwise fail to properly deliver the heartbeat packets between nodes, leading to a situation where all nodes believe they are master for certain VIPs. There is more detail on debugging switch issues with CARP in the section called “High Availability Troubleshooting”. A common misconception is that the failover is somehow signaled over the sync interface, but that

is incorrect. CARP happens everywhere CARP VIPs are defined, as that's the only way to reliably determine if a given link has failed.

### Note

Because each member of the CARP group must have an IP address in a subnet, plus the CARP IP address, at least three available IP addresses are required for each interface, and more IP addresses for additional group members. This also applies to your WAN interface, so be sure you have at least three available routable IP addresses from your ISP. The smallest routable block that includes 3 IP addresses is a /29, which has 8 addresses (6 usable).

# pfsync Overview

pfsync enables the synchronization of the firewall state table between the master and secondary firewalls. Changes on the primary's state table are sent out on the network to the secondary firewall(s), and vice versa. This uses multicast by default, though an IP address can be defined in the pfSense interface to force unicast updates for environments with only two firewalls where multicast traffic will not function properly (some switches block or break multicast). You can use any active interface for sending pfsync updates, however we recommend utilizing a dedicated interface for security and performance reasons. pfsync does not support any sort of authentication, so if you use anything other than a dedicated interface, it is possible for any user with local network access to insert states into your secondary firewall. 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 your network.

The benefit of pfsync is you can fail over without losing your state table, which allows for seamless failover. In some environments, you won't notice the difference between failing over statefully and losing state during failover. In other networks, it can cause a significant but brief network outage.

The pfsync settings *should be enabled* on all nodes participating in state synchronization, slave nodes included.

# 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 has changed to accommodate additional functionality. If you are upgrading from pfSense 1.2.x to 2.x or higher, the underlying OS changed from FreeBSD 6.x or 7.x to FreeBSD 8.x and includes a newer pfsync. Always check the upgrade guide linked in all release announcements before upgrading to see if there are any special considerations for CARP users.

### Note

On pfSense 2.1, an automatic firewall rule for pfsync was removed. The documentation has always recommended adding your own rule, and the automatic rule could be too permissive in some cases when pfsync must happen on an interface that is shared with other traffic. Ensure that your chosen sync interface has a rule to pass pfsync traffic before upgrading to 2.1.

# pfSense XML-RPC Config Sync Overview

pfSense's configuration synchronization allows you to make most configuration changes on only the primary firewall, which then replicates those changes over to the secondary automatically. The areas supported by this are users and groups, authentication servers, certificates, firewall rules, firewall schedules, aliases, NAT, IPsec, OpenVPN, DHCP, Wake on LAN, routes and gateways,

load balancer, virtual IPs, traffic shaper (queues, limiters, and layer 7), DNS forwarder, and captive portal. Other settings must be individually configured on the secondary firewall as needed, though the synchronization covers most if not all of what you will routinely change. Configuration synchronization should use the same interface as your pfsync traffic.

The XML-RPC settings should *only* be enabled on the primary node, all other nodes should have these settings *disabled*.

# Config sync and upgrades

As with pfsync, some care may need to be taken when upgrading a firewall cluster with config sync enabled when moving from 1.2.x to 2.x. You'll upgrade either the primary or the secondary first, leaving the other on 1.2.3 until testing is complete. Whether to choose the primary or the secondary depends on your preference, though there are additional considerations. Historically we've recommended upgrading the secondary first, verifying it functions as desired, and then upgrading the primary. However, with 2.0 the opposite may be preferable. 1.2.3 does not check the version it's syncing its config to, so it will overwrite pieces of a 2.x config with the old config structure which isn't correct for 2.x. This means upgrading the primary may be preferable as 2.x will not sync its config to 1.2.3 to avoid the problems that happen when syncing the wrong configuration version.

If you upgrade the secondary first and the primary is still running a 1.x version, take the IP, username and password out of Firewall → Virtual IPs, CARP Settings tab on the primary until the primary is upgraded to 2.1. Once you have upgraded the primary to 2.1 you can fill in the options again. Note that on 2.1 the location for these settings has changed to System → High Avail. Sync.

# Example Redundant Configuration

This section describes the steps in planning for and configuring a simple three interface HA configuration. The three interfaces are LAN, WAN, and pfsync. 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.

# Determine IP Address Assignments

First you need to plan your IP address assignments. A good strategy is to use the lowest usable IP in the subnet as the CARP IP, the next subsequent IP as the primary firewall's interface IP, and the next IP as the secondary firewall's interface IP. You can assign these as desired, so choosing a scheme that makes the most sense to you is recommended.

## WAN Addressing

The WAN addresses will be selected from those assigned by your ISP. For the example in Table 25.1, “WAN IP Address Assignments”, the WAN of the HA pair is on a private network, and the addresses 10.0.66.10 through 10.0.66.12 will be used as the WAN IPs.

### Table 25.1. WAN IP Address Assignments

| IP Address | Usage |
|---|---|
| 10.0.66.10 | CARP shared IP |
| 10.0.66.11 | Primary firewall WAN IP |
| 10.0.66.12 | Secondary firewall WAN IP |

## LAN Addressing

The LAN subnet is 192.168.1.0/24. For this example, the LAN IPs will be assigned as shown in Table 25.2, “LAN IP Address Assignments”.

### Table 25.2. LAN IP Address Assignments

| IP Address | Usage |
|---|---|
| 192.168.1.1 | CARP shared IP |
| 192.168.1.2 | Primary firewall LAN IP |
| 192.168.1.3 | Secondary firewall LAN IP |

## pfsync Addressing

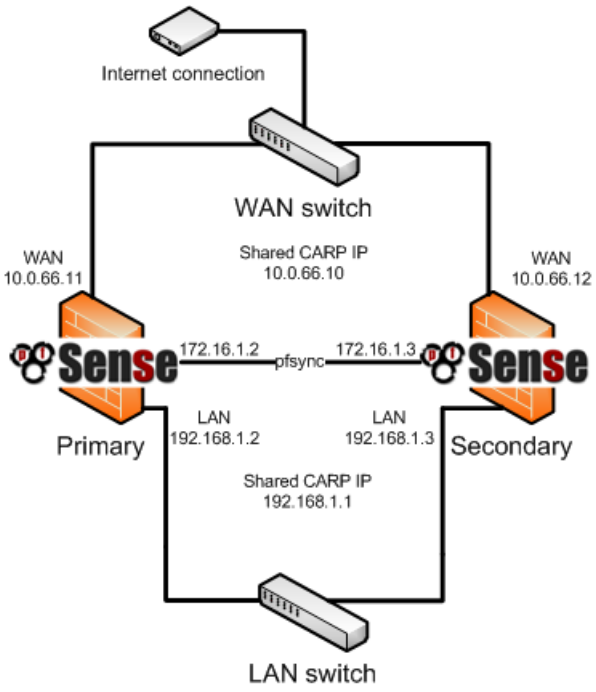
There will be no shared CARP IP on this interface because there is no need for one. These IPs are used only for communication between the firewalls. For this example, I will use 172.16.1.0/24 as the pfsync subnet. Only two IPs will be used, but I will use a /24 to be consistent with the other internal interface (LAN). For the last octet of the IP addresses, I chose the same last octet as that firewall's LAN IP for consistency.

### Table 25.3. pfsync IP Address Assignments

| IP Address | Usage |
|---|---|
| 172.16.1.2 | Primary firewall LAN IP |
| 172.16.1.3 | Secondary firewall LAN IP |

In Figure 25.1, “Example HA network diagram” you can see 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 pfsync 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 the section called “Layer 2 Redundancy”.

### Figure 25.1. Example HA network diagram




# Configure the primary firewall

First we will get everything functioning as desired on the primary, then the secondary will be added. Leave the secondary firewall turned off until you get to that point.

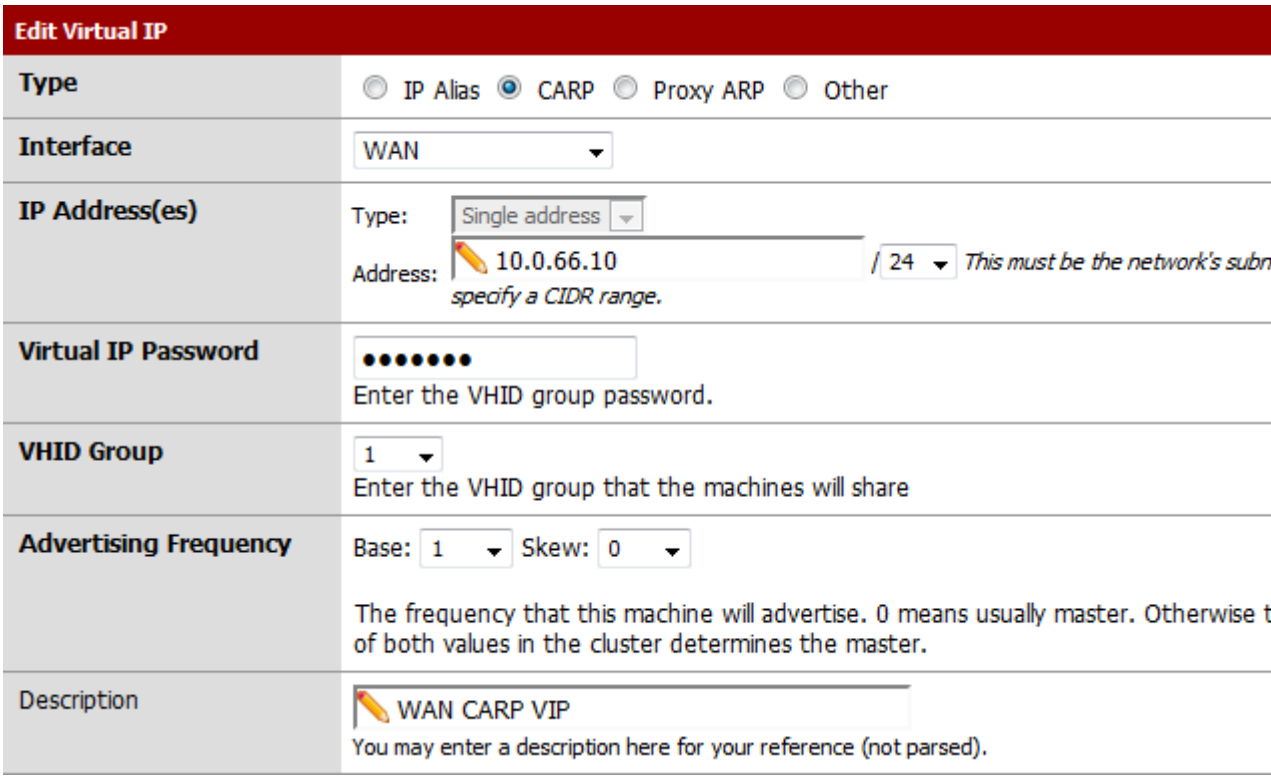
## Installation, interface assignment and basic configuration

Go through the installation and interface assignment no differently than you would for a single install. Assign the previously designated IP address to the LAN interface, and log into the web interface to continue. Go through the setup wizard, selecting your timezone, configuring the static IP previously designated for the primary firewall on the WAN, and setting your admin password. Continue to the next step after completing the setup wizard (refer back to the section called “Setup Wizard” if needed).

## Configuring the CARP Virtual IPs

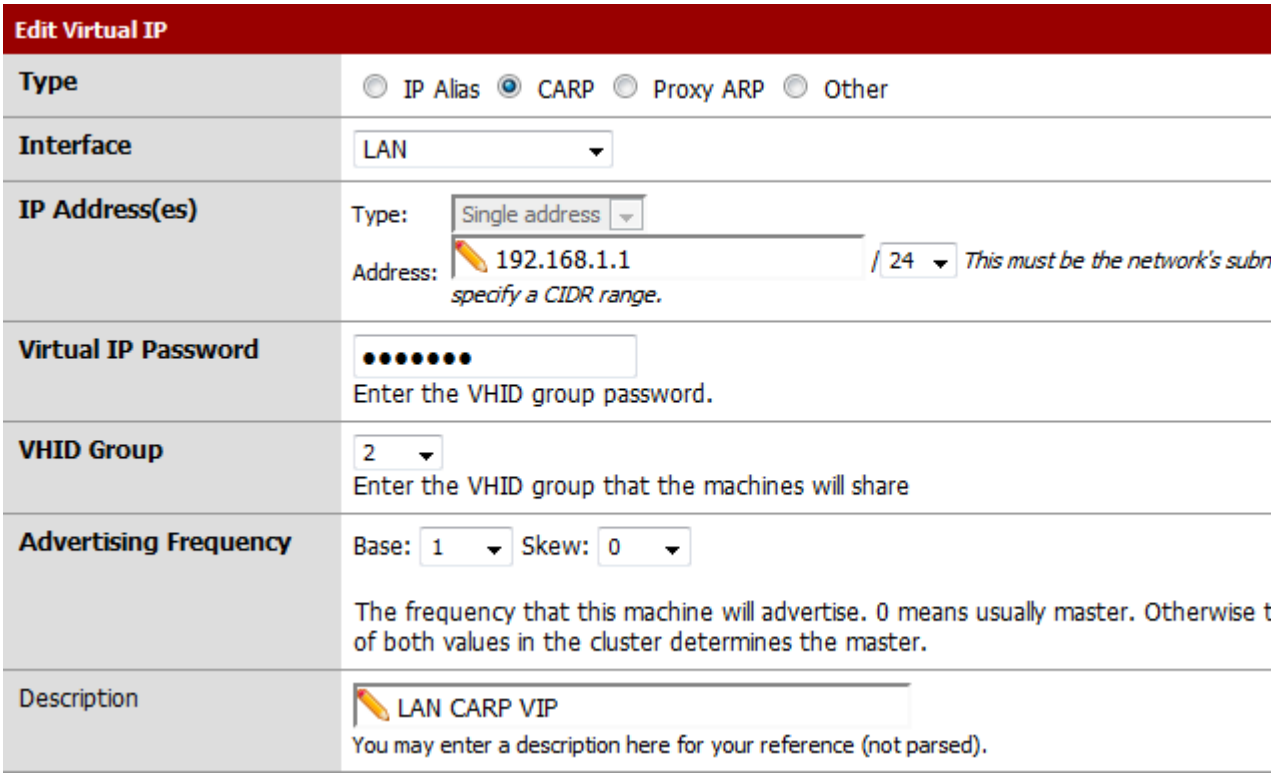
Browse to Firewall → Virtual IPs and click to add your first CARP VIP. The virtual IP editing screen will be displayed, as seen in Figure 25.2, “WAN CARP IP”

**Figure 25.2. WAN CARP IP**



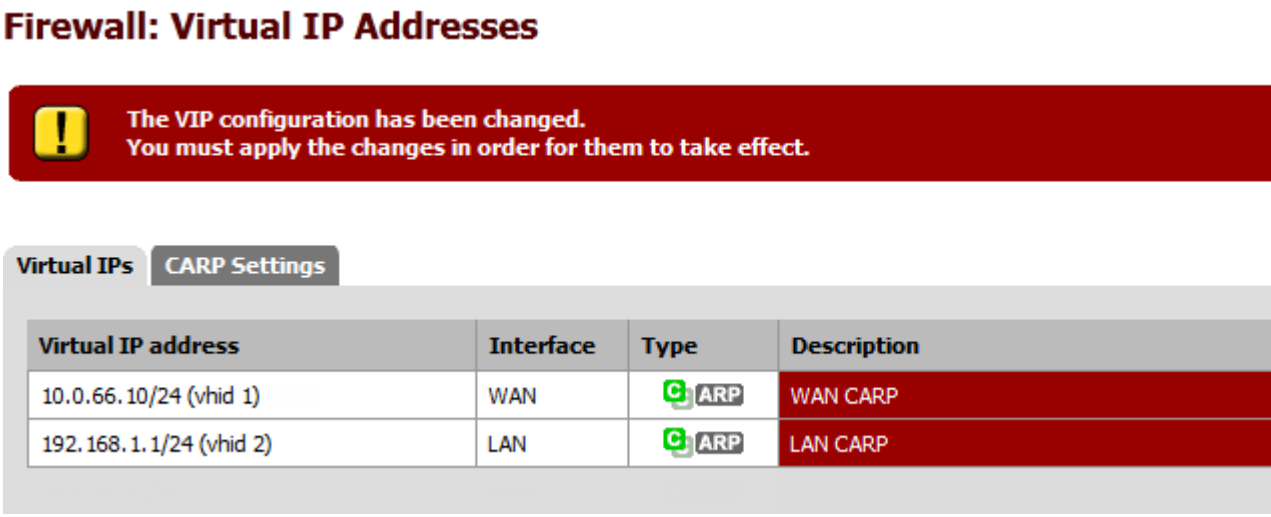

For the Type, select **CARP**. The Interface should be set to **WAN**. For the IP Address, enter in the shared WAN IP address chosen earlier. In this example, it is *10.0.66.10*. Ensure that the subnet mask for that address matches the subnet mask for the interface, **24**. The Virtual IP Password can be whatever you like, and as long as all your systems use pfSense with its configuration synchronization, you never need to know this password as it will automatically synchronize to your secondary firewall. You can generate a random password using a password generation tool, or bang randomly on the keyboard to create one. Each CARP IP on a pair of firewalls must use a unique VHID group (Virtual Host ID), and it also must be different from any VHIDs in active use on any directly connected network interface if CARP or VRRP is also present on other routers or firewalls on your network. If you have no other CARP or VRRP traffic present on your network, you may start at **1**. Otherwise, set it to the next available VHID on your network. The Advertising Frequency value for Skew should be set according to this machine's role in the group. Since this one will be master, it should be set to **0**. On the backup system, this should be **1** or higher. The Base can typically be left at the default of 1, but in some networks, such as those with high latency between nodes, a higher base can make the failover process more forgiving, though failover will take longer to occur. For the Description, enter something relevant such as **WAN CARP IP**. Click Save when finished.

**Figure 25.3. LAN CARP IP**




Now click to add another Virtual IP for the LAN (Figure 25.3, "LAN CARP IP"). This time, set Type to **CARP**, Interface to **LAN**, and IP Address to the shared LAN IP, **192.168.1.1** and the CIDR to **24**. This Virtual IP Password is for a different IP group, so it does not have to match the one for WAN, and again you will never need to know this password. The VHID should be different from that of the WAN CARP IP, typically it is set one number higher, in this case **2**. Again, since this system is master the Advertising Frequency Skew should be **0**. For the Description, enter **LAN CARP IP** or something similarly descriptive. Click Save when finished.

**Figure 25.4. Virtual IP list**



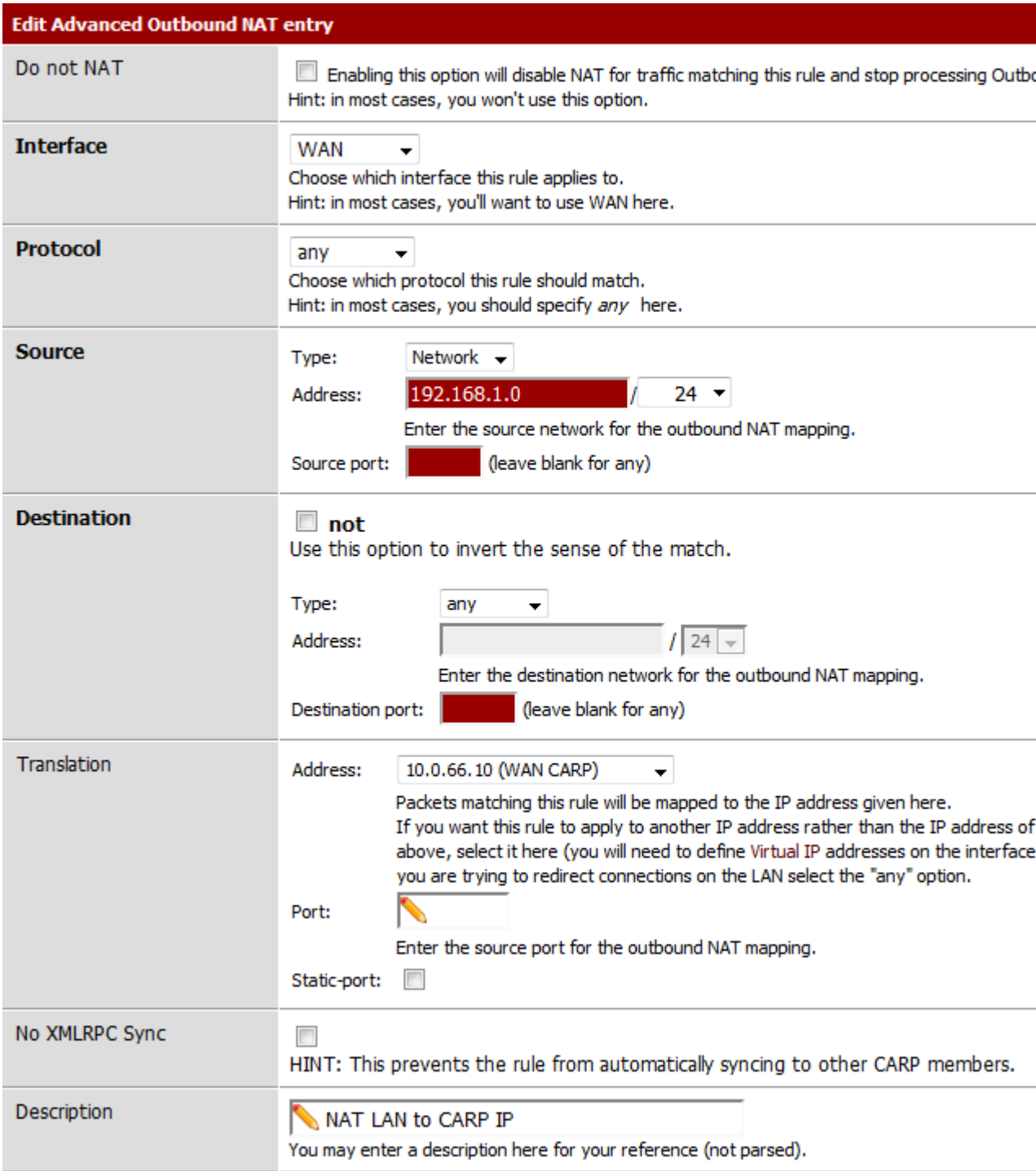

After saving the LAN CARP IP, you will see both VIPs in the list, as in Figure 25.4, "Virtual IP list". Click Apply Changes and then both CARP IPs will be active.

## 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. Browse to Firewall → NAT, and click the Outbound tab. Select the option to enable Manual Outbound NAT (Advanced Outbound NAT), then click Save.

A set of rules will appear that are the equivalent rules to those in place for Automatic Outbound NAT, for your convenience. These rules will do things like NAT your LAN traffic to the WAN IP. You can adjust these rules to work with the CARP IP address instead. Click the to the right of the rule. In the Translation section, select the WAN CARP IP address from the Address drop-down. Change the Description to mention that this rule will NAT LAN to the WAN CARP. For reference, you may compare your outbound NAT rule settings to those in Figure 25.5, “Outbound NAT Entry”

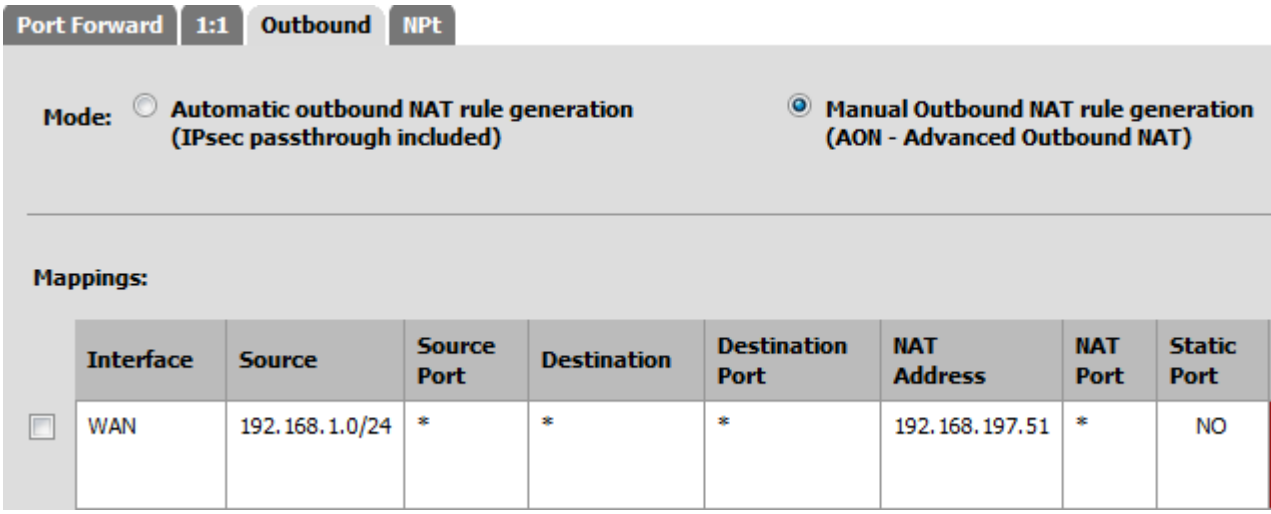
**Figure 25.5. Outbound NAT Entry**




After you click Save on the NAT rule, and then click Apply Changes, new connections leaving the WAN will now be translated to the CARP IP. You can confirm this with a web site that displays the IP address from which is it being accessed, such as http://www.pfsense.org/ip.php .

You should also see the properly setup outbound NAT rule in the list, as in Figure 25.6, “Advanced Outbound NAT Configuration”.

**Figure 25.6. Advanced Outbound NAT Configuration**

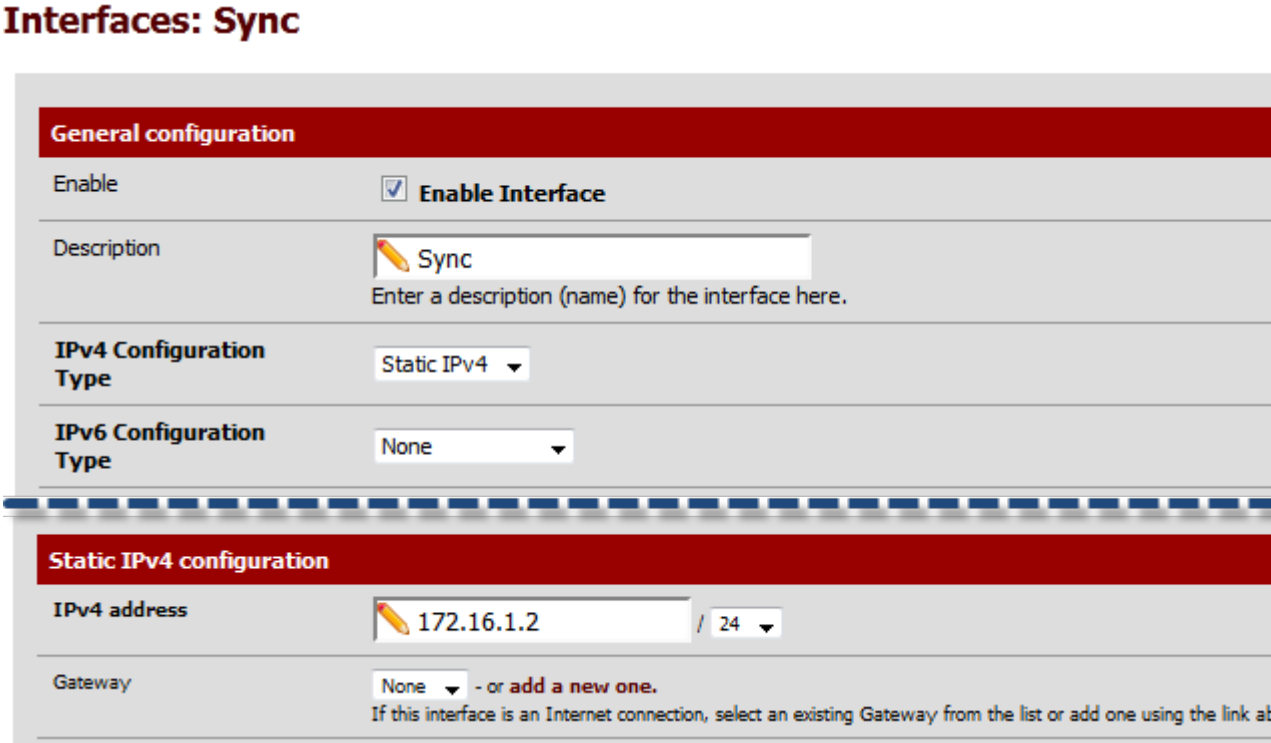



## Configure pfsync

The next task is to configure the pfsync interface that will be the line of communication between the primary and backup firewall. Navigate to Interfaces → OPT1 to set up the sync interface. If you do not have an OPT1 interface yet, you will need to assign it under Interfaces → (assign) (see the section called “Assign interfaces”).

Only a few options need to be set, as shown in Figure 25.7, “pfsync Interface Configuration”. The interface needs to be enabled, and it would help to use **Sync** for its name. It should be set for a static IP, and given the address decided upon earlier for the primary side of pfsync, **`172.16.1.2/24`**.

**Figure 25.7. pfsync Interface Configuration**




When you have finished entering the information for the Sync interface, click Save.

The Sync interface will also need a firewall rule to allow traffic from the backup. Go to Firewall → Rules, and click the Sync tab. Add a new firewall rule that will allow traffic of any protocol from the Sync subnet to any destination. Since this will only be a direct private connection with a crossover cable, it is safe to allow all traffic between the sync peers.

## Modifying the DHCP Server

If pfSense is acting as a DHCP server, you need to instruct it to assign a CARP IP as the gateway IP. Otherwise pfSense will use its default behavior of assigning the IP configured on that interface as the gateway. That IP is specific to the primary firewall, so you need to change to a CARP IP for failover to work for your DHCP client systems.

Browse to Services → DHCP Server. Change the Gateway field to **`192.168.1.1`**, the shared CARP LAN IP. Set the Failover peer IP to the actual LAN IP of the backup system, **`192.168.1.3`**. This will allow the DHCP service on both systems to maintain a common set of leases.

Save, then Apply Changes.

# Configuring the secondary firewall

Next the interfaces, IP addresses, and firewall rules on the secondary need to be configured.

## Interface assignment and IP addressing

Before plugging in the WAN, LAN, or Sync interfaces, power on the system and go through the installation and interface assignment as you did for the primary firewall. Set the LAN IP from the console to the previously designated backup LAN IP of **`192.168.1.3`**, set the DHCP settings the same as the primary, and then it should be safe to plug in the network connections.

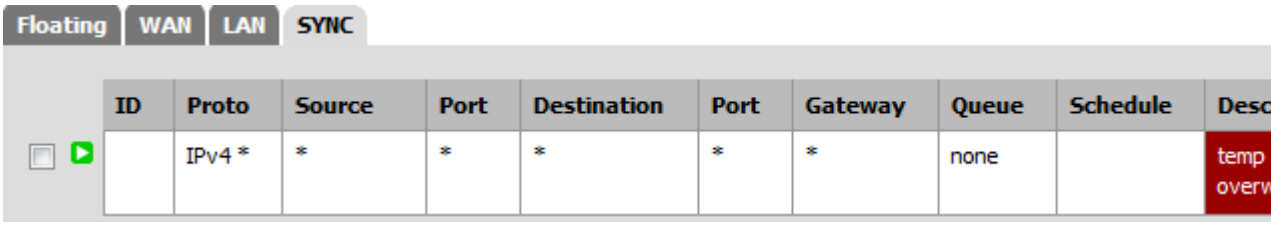
You should then login to the web interface and go through the setup wizard, just as was done on the primary. Configure the WAN IP, and set the admin password to the same password as that on the primary.

You will also need to setup the sync interface as in the section called “Configure pfsync”, but with the IP address chosen for the backup system

## Firewall rules

You will need a temporary firewall rule to allow the initial configuration sync to happen. Go to Firewall → Rules, and click the Sync tab. Add a new firewall rule that will allow traffic of any protocol from any source to any destination. Put "temp" in the description so you can be sure that it has been replaced later. The rule should look like Figure 25.8, “Firewall rule on Sync interface”

**Figure 25.8. Firewall rule on Sync interface**




# Setting up configuration synchronization

On the backup firewall, go to System → High Avail. Sync. Check Synchronize States, pick Sync as the Synchronize Interface, and for the pfsync Synchronize Peer IP, enter the IP address for the primary

system's pfsync interface, **172.16.1.2**. Click Save when finished. Do not set any other values on this page.

The final step is to configure the configuration synchronization between the primary and backup. On the master firewall, go to System → High Avail. Sync.

Check Synchronize States, and pick **Sync** as the Synchronize Interface. For the pfsync Synchronize Peer IP, enter the IP address for the backup system's pfsync interface, **172.16.1.3**. Enter the backup system's pfsync IP again in Synchronize Config to IP. Then, enter the WebGUI admin password in the Remote System Password box. Check all of the remaining boxes on the screen to make sure all possible configuration areas will be synchronized. Click Save when finished.

When the synchronization settings are saved on the primary, it will automatically copy the settings from the primary to the backup for each selected option on the High Avail. Sync page. This includes the proper outbound NAT settings for HA, the firewall rules for the Sync interface, and even the CARP VIPs. Within 30 seconds, the initial configuration sync should have finished.

The DHCP server settings are synchronized, and the system is smart enough to adjust the failover IP as needed to make sure that DHCP failover works, so long as you set the DHCP settings as described in the section called “Modifying the DHCP Server”.

If the settings synchronized from the primary to the backup, then you know that the sync interface is connected and working properly. If not, you can go to Diagnostics → Ping, pick the Sync interface, and attempt to ping the Sync IP address of the opposing system. If that does not work, check if you need to use a crossover cable and/or have a link light on the Sync interface of both systems.

The HA pair will now be active, but you will still need to check the status and test that failover is working properly. Skip down to the section called “Verifying Failover Functionality” for the rest.

### Note

You should **not** setup Synchronize Config to IP from the backup firewall to the master firewall. There are protections that should prevent this synchronization loop from causing harm, but it will clutter your logs with error messages and should never be configured this way. You **should** setup Synchronize States on both the master and the slave, as described in this section.

# Multi-WAN with HA

You can also deploy HA for firewall redundancy in a multi-WAN configuration, as long as all your WAN interfaces have at least 3 static IPs each. 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 IPs will be used on each WAN. Each firewall needs an IP, plus one CARP IP for Outbound NAT, plus one for a 1:1 NAT that will be used for an internal mail server in the DMZ segment.

## WAN and WAN2 IP Addressing

Table 25.4, “WAN IP Addressing” and Table 25.5, “WAN2 IP Addressing” show the IP addressing for both WANs. In most environments these will be public IPs.

### Table 25.4. WAN IP Addressing

| IP Address | Usage |
|---|---|

| | |
|---|---|
| 10.0.66.10 | Shared CARP IP for Outbound NAT |
| 10.0.66.11 | Primary firewall WAN IP |
| 10.0.66.12 | Secondary firewall WAN IP |
| 10.0.66.13 | Shared CARP IP for 1:1 NAT |

### Table 25.5. WAN2 IP Addressing

| IP Address | Usage |
|---|---|
| 10.0.64.90 | Shared CARP IP for Outbound NAT |
| 10.0.64.91 | Primary firewall WAN2 IP |
| 10.0.64.92 | Secondary firewall WAN2 IP |
| 10.0.64.93 | Shared CARP IP for 1:1 NAT |

## LAN Addressing

The LAN subnet is 192.168.1.0/24. For this example, the LAN IPs will be assigned as follows.

### Table 25.6. LAN IP Address Assignments

| IP Address | Usage |
|---|---|
| 192.168.1.1 | CARP shared LAN IP |
| 192.168.1.2 | Primary firewall LAN IP |
| 192.168.1.3 | Secondary firewall LAN IP |

## DMZ Addressing

The DMZ subnet is 192.168.2.0/24. For this example, the LAN IPs will be assigned as follows in Table 25.7, “DMZ IP Address Assignments”.

### Table 25.7. DMZ IP Address Assignments

| IP Address | Usage |
|---|---|
| 192.168.2.1 | CARP shared DMZ IP |
| 192.168.2.2 | Primary firewall DMZ IP |
| 192.168.2.3 | Secondary firewall DMZ IP |

## pfsync Addressing

There will be no shared CARP IP on this interface because there is no need for one. These IPs 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 IPs 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 25.8. Sync IP Address Assignments

| IP Address | Usage |
|---|---|
| 172.16.1.2 | Primary firewall LAN IP |
| 172.16.1.3 | Secondary firewall LAN IP |

# NAT Configuration

The NAT configuration when using HA is the same as without it, though you need to use only CARP VIPs, or public IPs in a subnet routed to one of your CARP IPs to ensure these addresses are always accessible. See Chapter 11, *Network Address Translation* for more information on NAT configuration.

# Firewall Configuration

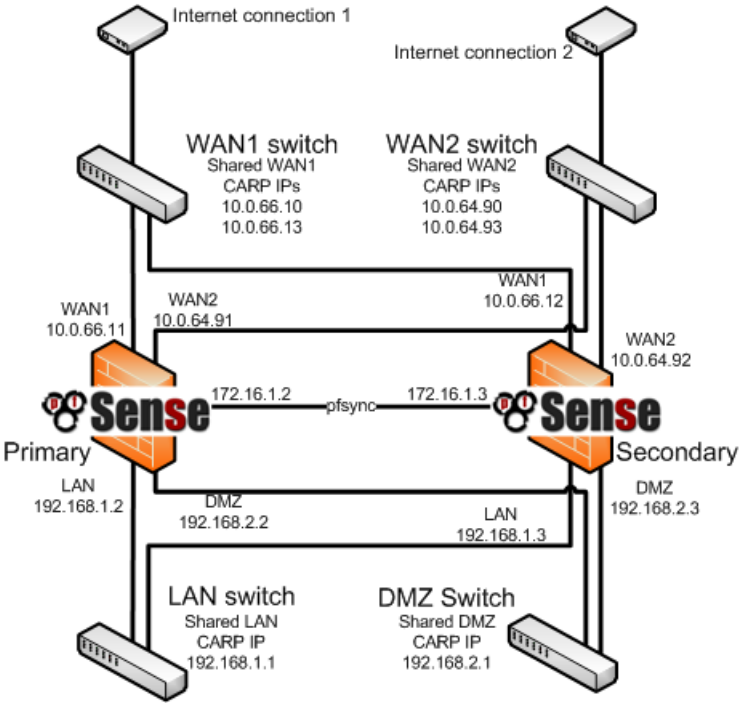
With Multi-WAN you need a policy for the local network to route to the default gateway otherwise when you attempt to send traffic to the CARP address it will instead go out a secondary WAN connection.

You need to add a rule 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 connections. The destination for this rule should 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 25.9, “Diagram of Multi-WAN HA with DMZ”.

**Figure 25.9. Diagram of Multi-WAN HA with DMZ**




# Verifying Failover Functionality

Since using HA is about high availability, it should be thoroughly tested before being placed into production. 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 the section called “High Availability Troubleshooting”.

# Check CARP status

On both systems, navigate to Status → CARP (failover). The primary should show MASTER for the status of all CARP VIPs. The backup system should show BACKUP as the status. If the backup system instead shows DISABLED, click the Enable CARP button, and then refresh the Status → CARP (failover) page. It should now show up correctly.

# Check Configuration Replication

Navigate to key locations on the backup router, such as Firewall → Rules and Firewall → NAT and ensure that rules created only on the primary system are being replicated to the backups.

If you followed the example earlier in this chapter, you should see that your "temp" firewall rule on the pfsync interface has been replaced by the rule from the primary.

# Check DHCP Failover Status

If you have configured DHCP failover, its status can be checked by going to 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 25.10, “DHCP Failover Pool Status”.

**Figure 25.10. DHCP Failover Pool Status**

| Failover Group | My State | Since | Peer State | Since |
|---|---|---|---|---|
| dhcp_lan (LAN) | normal | 2013/07/13 22:18:06 | normal | 2013/06/20 19:43:1 |

# Test CARP Failover

Now for the real failover test. Before starting, make sure that you can surf from a client behind the CARP pair with both pfSense firewalls online and running. Once that is confirmed to work, it would be an excellent time to make a backup.

For the actual test, unplug the primary from the network or shut the system down temporarily. You should be able to keep surfing the Internet through the backup router. Check Status → CARP (failover) again on the backup and it should now report that it is MASTER for the LAN and the WAN CARP VIPs.

Now bring the primary system back online and it should regain its role as MASTER, and the backup system should demote itself to BACKUP once again, and Internet connectivity should still work properly.

You should test the HA pair in as many failure scenarios as possible. Some other individual tests may include:

- Unplug the WAN or LAN cable
- Pull the power plug of the primary
- Disable CARP on the primary
- Test with each system individually (power off backup, then power back on and shut down the primary)
- Download a file or try streaming audio/video during the failover

- Try a continuous ping to an Internet host during the failover

# Providing Redundancy Without NAT

As mentioned earlier, only CARP VIPs provide redundancy and they can only be used in conjunction with NAT. You can also provide redundancy 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 co-location environments.

# Public IP Assignments

You will need at least a /29 public IP block for the WAN side of pfSense, 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 you need at least one CARP VIP on the WAN side.

The second public IP subnet will be routed to one of your CARP VIPs by your ISP, co-location provider, or your upstream router if you control that portion of the network. 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 /23 public IP subnet will be used and it will be subnetted into two /24 networks.

# Network Overview

The example network depicted here is a co-location environment consisting of two pfSense installs 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 is where your network connects to the upstream network, either your ISP, co-location provider, or your upstream router.

## WEB Network

The WEB segment in this network uses the "LAN" interface but renamed. It contains web servers, so we've named it WEB but you could call it DMZ, SERVERS, or anything you wish. pfSense 2.x no longer restricts you from renaming the LAN interface.

## 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 should never require direct access from the Internet, and hence are less subject to compromise than your web servers.

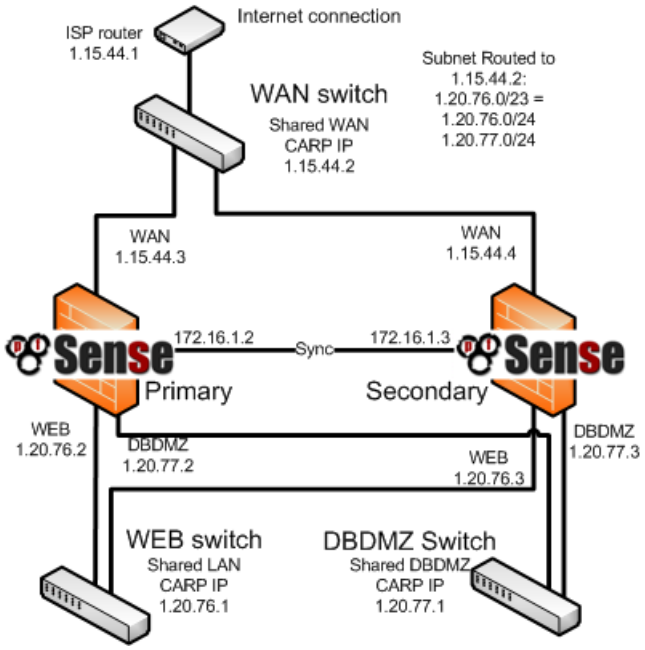
## Sync Network

The Sync network in this diagram is used to replicate pfSense configuration changes via config sync 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 25.11, “Diagram of HA with Routed IPs” illustrates this network layout, including all routable IP addresses, the WEB network, and the Database DMZ.

**Figure 25.11. Diagram of HA with Routed IPs**




### Note

Segments containing database servers typically do not need to be publicly accessible, and hence would more commonly use private IP subnets, but the example illustrated here can be used regardless of the function of the two internal subnets.

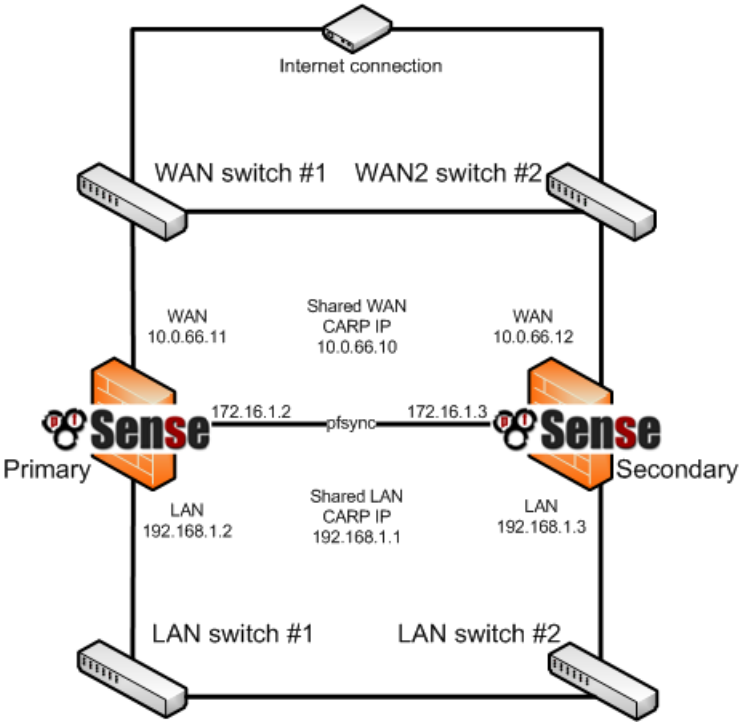
# Layer 2 Redundancy

The diagrams earlier in this chapter did not describe layer 2 (switch) redundancy, to avoid throwing too many concepts at readers simultaneously. Now that you have an understanding of hardware redundancy with pfSense, this section covers the layer 2 design elements you should consider when planning a redundant network. This chapter assumes a two system deployment, though this scales to as many installations as you require.

If both your redundant pfSense systems 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).

The Routed IPs diagram is network-centric, not showing the switch infrastructure. The Figure 25.12, “Diagram of HA with Redundant Switches” illustrates how that environment looks with a redundant switch infrastructure.

**Figure 25.12. Diagram of HA with Redundant Switches**




# Switch Configuration

When using multiple switches, you should interconnect them. As long as you have a single connection between the two switches, and do not 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. You will need a managed switch that 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.

In pfSense 2.0 and higher, support also exists for `lagg(4)` link aggregation and link failover interface which will allows you to have multiple network interfaces plugged into one or more switches for increased fault tolerance. See the section called “LAGG (Link Aggregation)” for more information on configuring link aggregation.

# Host Redundancy

It is more difficult to obtain host redundancy for your 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 you may be able to run CARP on a set of servers so that they would be redundant as well. Providing host redundancy is more specific to the capabilities of your switches and your server operating system, 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 you are hoping 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:

- Each redundant segment should have isolated power.

    - Redundant systems should be on separate breakers.
    - 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, T1, 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. Why buy one when you can buy two for twice the price?
- 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 using in production. It requires a lot of manual intervention. The details of the process can be found in the section called “High Availability”.

# Using IP Aliases to Reduce Heartbeat Traffic

If you have a large number of CARP VIPs on a segment, this can lead to a lot of multicast traffic. One heartbeat per second, per CARP VIP, is sent. To reduce this traffic, you can select one CARP VIP for an interface to be the "main" VIP. Then, you can 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.

Doing 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 the configuration sync, however, IP alias VIPs set to use CARP interfaces do synchronize.

# Using IP Aliases to handle CARP on Multiple Subnets on a Single Interface

If you need to have multiple subnets on a single interface to use with HA, you can do this by using IP Aliases. As with the main interface IPs, each firewall must have an IP address inside the additional subnet, for a total of at least three IPs per subnet. You will need to add separate IP alias entries 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 you have added the IP Alias VIP to both nodes to gain a foothold in the new subnet, you may then add CARP VIPs using IP addresses from there.

# 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 you still have issues after reading this section, there is a dedicated CARP/VIPs board on the pfSense Forum [http://forum.pfsense.org/index.php/board,36.0.html].

Before going much farther, 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 backup members, and watch for any places where the configuration should be different on the backups. Be sure to check the CARP status (the section called “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 you create on a given interface/broadcast domain. With a single HA pair, our 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 you also must ensure there are no conflicts with VRRP VHIDs, such as if your provider or another router on your network is using VRRP.

The best way around this is to use a unique set of VHIDs. If you are on a known safe private network, start numbering at 1. If you are on a network where VRRP or CARP are conflicting, you may have to 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

You must use the real subnet mask 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 IP 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 might pop up in the system logs, some more worrisome than others.

If CARP is not working properly when you see this error, 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 your 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. You may need to disable CARP and monitor the network with **tcpdump** (Chapter 30, *Packet Capturing*) to check for other CARP or CARP-like traffic, and adjust your VHIDs appropriately.

If CARP is working properly, and you see this message 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, backup shows BACKUP for status).

# Both Systems Appear as MASTER

This will happen if the backup cannot see the CARP advertisements from the master. Check for firewall rules, connectivity trouble, switch configurations. Also check the system logs for any relevant errors that might lead to a solution. If you are seeing this in a Virtual Machine (VM) Product such as ESX, see the section called “Issues inside of Virtual Machines (ESX)”.

# Master system is stuck as BACKUP

In some cases, this is may happen normally for a short period after a system comes back to life. 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. You can check this from the shell or Diagnostics → Command.

```
# ifconfig lan_vip1
lan_vip1: flags=49<UP,LOOPBACK,RUNNING> mtu 1500
 inet 10.0.66.10 netmask 0xffffff80
 carp: BACKUP vhid 1 advbase 1 advskew 240
```

In that case, you should isolate that firewall and perform further hardware testing.

# 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 you have a Virtual Distributed Switch, you can make a port group for the firewall interfaces with promiscuous mode enabled, and a separate non-promiscuous port group for your 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 you use VDS (Virtual Distributed Switches) 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 you create a new VDS 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 you enable port mirroring on a VDS, it will break promiscuous mode. To fix it, you must disable promiscuous mode, then re-enable promiscuous mode.

## ESX Client Port Issues

If you have a physical HA cluster connected to a switch with an ESX box using multiple ports on the ESX box (lagg group or similar), and you find that only certain devices/IPs are reachable by the target VM, then you may need to adjust the port group settings 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 IP from the target VM when you can reach the "real" IP of the primary firewall.
- Port forwards or other inbound connections to the target VM work from some IPs and not others.

## ESX Physical NIC Failure Fails to Trigger Failover

Because CARP's self-demotion relies on the loss of link on a switch port, you might find that if you have the primary and secondary on separate ESX units, and the primary unit loses a switch port link and does not expose that to the VM, that 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.

We're investigating possible long-term solutions for this in the future to work around this VM limitation.

## KVM+Qemu Issues

Be sure to use e1000 NICs (`em(4)`), not the `ed(4)` NICs or your 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 you are using the switch on the back of a modem/CPE, 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 master 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.
- You must permit traffic to the WebGUI port on the interface which handles the synchronization traffic.
- The pfsync interface must be enabled and configured on all nodes.
- Remove **all** special characters from every description that you are syncing: NAT rules, Firewall rules, Virtual IPs, etc. This should no longer pose a problem, but should you have difficulties it is a good thing to try.
- Verify that **only** the master sync node has the configuration synchronization options enabled.
- Ensure no IP address is specified in the Synchronize Config to IP on the backup node.
- Ensure the clocks on both nodes are current and are reasonably accurate.

# HA and Multi-WAN Troubleshooting

If you have trouble reaching CARP VIPs from when dealing with Multi-WAN, double check that you have a rule such as one mentioned in the section called “Firewall Configuration”

# Removing a CARP VIP

In pfSense 1.2.x, removing a CARP VIP required a reboot. That is no longer the case on pfSense 2.x. CARP VIPs may be removed as needed.

# Chapter 26. Services

The base install of pfSense comes along with a set of services which add some fundamental functionality and flexibility to the firewall system. As the name implies, the options found within control services that the router will provide to clients, or in the case of routing services, other routers as well. These services include providing DHCP addressing for IPv4 and IPv6, DNS resolution and Dynamic DNS, SNMP, UPnP and much more. This chapter covers the services available in the base system. There are many more services that can be added with packages, which will be covered later in the book.

# IPv4 DHCP Server

The DHCP server assigns IP addresses and related configuration options to client PCs on your network. It is enabled by default on the LAN interface, and with the default LAN IP of 192.168.1.1, the default scope range would be 192.168.1.100 through 192.168.1.199. In its default configuration, pfSense assigns its LAN IP as the gateway and DNS server if the DNS Forwarder is enabled. There are many options available to adjust in the WebGUI. The DHCP server has gained many extra abilities since pfSense 1.2.3, such as the ability to define any custom numbered option, the ability to define extra pools of IPs inside the interface's subnet, and the ability to set options specific for static mappings.

## Configuration

To alter the behavior of the DHCP server, go to Services → DHCP Server. From there you can alter the behavior of the DHCP server, along with static IP mappings and some related options like static ARP.

### Choosing an Interface

On the DHCP configuration page there is a tab for each interface with a static IP. Each interface has its own separate DHCP server configuration, and they may be enabled or disabled independently of one another. Before making any changes, ensure that you are looking at the tab for the correct interface.

### Service Options

The first setting on each tab tells pfSense whether or not to handle DHCP requests on that interface. To enable DHCP on the interface, check the Enable DHCP server on [name] interface box. To disable the service, uncheck that same box.

Normally, the DHCP server will answer requests from any client which requests a lease. In most environments this is normal and acceptable behavior, but in more restricted or secure environments this behavior is undesirable. With the Deny unknown clients option set, only clients with static mappings defined will receive leases, which is a more secure practice but is much less convenient. This option is per-pool, meaning that if you deny unknown clients in the default range, you can specify another pool of IPs that does not have that setting checked and it will assign them IPs out of that alternate pool.

**Note**

This will protect against low-knowledge users and people who casually plug in devices. Be aware, however, that a user with knowledge of your 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 switch ports which you know should not be in use.

The IP address for the interface being configured is also shown, along with its subnet mask. Underneath that line the available range of IP addresses for that subnet mask is printed, which may help determine what starting and ending addresses to use for the DHCP pool range.

## Address Range (DHCP Pool)

The two boxes for Range tell pfSense what will be the first and last address for use as a DHCP pool. 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 would be **192.168.1.100** to **192.168.1.199**. This range can be as large or as small as your network needs, but it must be wholly contained within the subnet for the interface being configured.

## Additional Pools

A new option in pfSense 2.1, the Additional Pools section lets you define extra pools of addresses inside of the same subnet. You can use these to setup a pool of IPs specifically for certain clients, or for overflow from a smaller original pool, or to simply split up the main pool into two smaller chunks with a GAP of non-DHCP IPs in the middle of what used to be the pool. You can use a combination of the MAC Address Control options to guide clients from the same manufacturer into a specific pool, such as IP phones.

To add a new pool, click and the screen will switch to the pool editing view, which looks mostly 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 here 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 how you can direct clients into or away from pools.

## WINS Servers

Two WINS Servers (Windows Internet Name Service) may be defined that will be passed on to clients. If you have one or more WINS servers available, 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 your setup. If you are using the DNS Forwarder built into pfSense 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 assigned to it under System → General Setup. If you wish to use custom DNS Servers instead of the automatic choices, fill in the IP addresses for up to two DNS servers here. (See the section called “Free Content Filtering with OpenDNS” for an example.) In networks with Windows servers, especially those employing Active Directory, it is recommended to use those servers for client DNS. When using the DNS forwarder in combination with CARP, specify the CARP IP on this interface here.

## Gateway

The Gateway option may also be left blank if pfSense is the gateway for your network. Should that not be 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 IP on this interface here.

## Domain Name

The Domain Name option specifies the domain name passed to the client to form its fully qualified hostname. If the Domain Name is left blank, then the domain name of the firewall it sent to the client. Otherwise, the client is sent this value to be used for their domain name.

## Domain Search List

Domain Search List controls the DNS search domains that are provided to the client via DHCP. If you have multiple domains and you use short hostnames on them, 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.

## DHCP Lease Times

The Default lease time and Maximum lease time control how long a DHCP lease will last. The default lease time is used when a client does not request a specific expiration time. If the client does specify how long it wants a lease to last, the maximum lease time setting will let you limit that to a reasonable amount of time. These values are specified in seconds, and the default values are 7200 seconds (2 hours) for the default time, and 86400 seconds (1 day) for the maximum time.

## Failover

If this system is part of a failover setup such as a CARP cluster, enter the Failover peer IP address. This should be the real IP address of the other system in this subnet, not a shared CARP address.

## Static ARP

The Enable Static ARP entries checkbox works similarly to denying unknown MAC addresses from obtaining leases, but takes it a step further in that it would also restrict any unknown machine from communicating with the pfSense router. This would stop would-be abusers from hardcoding an unused address on this subnet, circumventing DHCP restrictions.

### Note

When using static ARP, be careful to ensure that all systems that need to communicate with the router are listed in the static mappings list before activating this option, especially the system being used to connect to the pfSense WebGUI. Also be aware that this option may prevent people from hardcoding an IP 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. If you check this option, the lease times are converted to your local timezone when they are displayed on the lease view.

## Dynamic DNS

For Dynamic DNS settings, click the Advanced button to the right of that field. To enable this function, check the box and then fill in a domain name for the DHCP hostnames. If you are using pfSense's DNS forwarder, you may instead leave this option blank and configure the setting inside of the DNS forwarder setup.

### Note

On versions of pfSense 2.0.x, this feature was non-functional. It was corrected in pfSense 2.1.

## MAC Address Control

The MAC Address Control section lets you specify a list, comma separated, of MAC address prefixes or full address that should be allowed into a pool or denied from a pool. This can be used to guide clients into a specific pool of addresses so they get different DHCP options applied. For example, if all of your VoIP phones are from the same manufacturer and start with the MAC address `aa:bb:cc`, you can deny them from the main DHCP pool and allow them into an additional pool where they can be given a different gateway or DNS servers.

When crafting entries for this box, keep in mind that if you put a MAC address into the allow box, then all others will be denied except the MAC address specified in the allow box. If you specify a MAC address to deny, 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 you do not take that extra step to allow their MAC prefix in the additional pool, then other non-VoIP phone clients could receive IPs from this pool, which may lead to undesired behavior.

You can also use this to blacklist certain devices from getting any DHCP response. For example if you do not want any Example brand printers to get a DHCP address and the MAC addresses all start with `ee:ee:ee`, then placing that in the deny list of every pool (or just the main pool if you only have one) will prevent them from receiving an IP address.

## NTP Servers

To specify NTP Servers (Network Time Protocol Servers), click the Advanced button to the right of that field, and enter IP addresses for up to two NTP servers.

## TFTP Server

The value in the TFTP Server box, if desired, should 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

LDAP URI will send an LDAP server URI to the client if requested. This may also be referred to as DHCP option 95. It should take 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.

## Network Booting

To view the Enable Network booting settings, click the Advanced button to the right of that field. You may then check the box to turn on the feature, and then enter an IP address from which boot images are available, and a file name for the boot image. Both of these fields must be configured for network booting to work properly. You may also optionally specify a Root Path String to target a specific device as the client's root filesystem device, such as `iscsi:(servername):(protocol):(port):(LUN):targetname`.

## Additional BOOTP/DHCP Options

If the above options aren't enough, any other numeric DHCP option code can be sent to clients using the Additional BOOTP/DHCP Options controls. IANA maintains a list of all valid DHCP options [http://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xml]. To add a new option, click and enter the option Number, select the Type, and then enter the Value to be delivered to the clients.

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`*. |

For more information on which options take a specific type or format, see the linked list above from the IANA.

## Save Settings

After making these changes, be sure to click Save before attempting to create static mappings. The settings will be lost if you navigate away from this page without saving first.

## Static Mappings

Static DHCP mappings allow you to express a preference for which IP address will be assigned to a given PC, based on its MAC address. In 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. First, from this screen, click and you will be presented with a form for adding a static mapping. The other method is to add them from the DHCP leases view, which is covered later in this chapter.

Of the fields on this screen, only the MAC address is necessary. 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. There is a link beside the MAC address field that will copy the MAC address of the PC being used to access the WebGUI. This is provided as a convenience, versus obtaining the MAC address in other, more complicated, ways.

### Note

The 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 card 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"`**.

The IP address field is needed if this will be a static IP mapping instead of only informing the DHCP server that the client is valid. This IP address is really a *preference*, and 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 one from the general pool. For this reason, the pfSense WebGUI does not allow you to assign static IP mappings inside of your DHCP pool.

A Hostname may also be set, and it 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.

The Description is cosmetic, and available for your 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.

If you check ARP Table Static Entry, this entry will receive a static ARP entry in the OS tying this IP address to this MAC address. Note that if you use this 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.

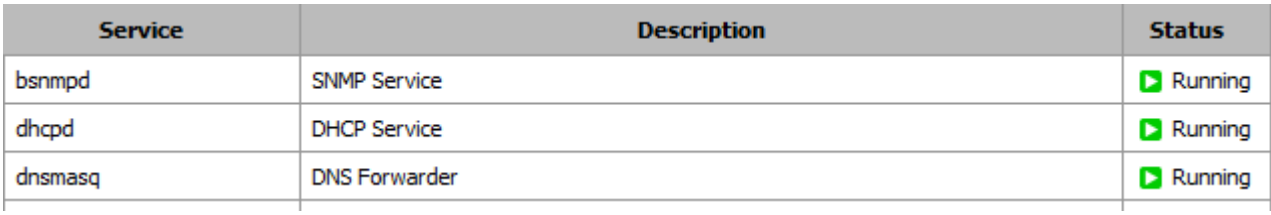
The remaining options available to set for this client are the same in behavior to the ones found earlier in this chapter for the main DHCP settings.

Click Save to finish editing the static mapping and return to the DHCP Server configuration page.

# Status

You will find the status of the DHCP server service itself under Status → Services. If it is enabled, its status should be shown as Running, as in Figure 26.1, “DHCP Daemon Service Status”. The buttons on the right side allow you to restart or stop the DHCP server service. Restarting should never be necessary as pfSense will automatically restart the service when configuration changes are made that require a restart. Stopping the service is also likely never necessary, as the service will stop when you disable all instances of the DHCP server.

**Figure 26.1. DHCP Daemon Service Status**



| Service | Description | Status |
|---|---|---|
| bsnmpd | SNMP Service | Running |
| dhcpd | DHCP Service | Running |
| dnsmasq | DNS Forwarder | Running |

# Leases

You can view the currently assigned leases at Status → DHCP leases. This screen shows the assigned IP address, the MAC address it is assigned to, the hostname (if any) that the client sent as part of the DHCP request, the beginning and end times of the lease, whether the machine is currently online, and whether the lease is active, expired, or a static registration.

## View inactive leases

By default, only active and static leases are shown, but you may see everything, including the expired leases, by clicking the Show all configured leases button. To reduce the view back to normal, click the Show active and static leases only button.

## Wake on LAN Integration

If you click on the MAC address, or the Wake on LAN button to the right of the lease, pfSense will send a Wake on LAN packet to that host. For more details about Wake on LAN, see the section called “Wake on LAN”.

### Add static mapping

To make a dynamic lease into a static mapping, click the to the right of the lease. This will pre-fill the MAC address of that host into the "Edit static mapping" screen. You'll need to add the desired IP address, hostname and description and click **Save**. Any existing leases for this MAC address will be cleared out of the leases file when saving the new entry.

### Delete a lease

While viewing the leases, you may delete an inactive or expired lease manually by clicking the button at the end of a 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 how it works for 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

One quirk of using DHCPv6 is that not all clients support it. 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. You will have to consider your address tracking and operating system support requirements when deciding how to allocate IPv6 addresses to clients on your network.

Many operating systems such as Windows, OSX, FreeBSD, Linux, and their cousins contain DHCPv6 clients that are capable of obtaining addresses as expected via DHCPv6. Many lightweight or mobile operating systems such as Android and iOS 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, you locate a router through 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 your 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, you must at least select a mode from the Router Advertisements drop-down list.

## Router Advertisement Modes

The modes for the RA daemon not only control the services offered by pfSense and announce the firewall as an IPv6 router on the network, but they also direct clients on how to obtain addresses.

| | |
|---|---|
| Disabled | In the Disabled mode, the RA demon is disabled and will not run. IPv6 gateways must be entered manually on any client hosts. |
| Router Only | In Router Only mode, this firewall will send out RA packets that simply advertise that this firewall is capable of being an IPv6 router. DHCPv6 is disabled in this mode. |
| Unmanaged | In Unmanaged mode, the firewall will send out RA packets and clients are directed to assign themselves IPs within this interface's subnet using SLAAC. DHCPv6 is disabled in this mode. |
| Managed | In Managed mode, the firewall will send out RA packets and addresses are only assigned to clients by DHCPv6. |
| Assisted | In Assisted mode, the firewall will send out RA packets and addresses can be assigned to clients by DHCPv6 or SLAAC. |

## Router Priority

If there are multiple IPv6 routers on the same network segment, they can indicate to clients in which order they should be used, should one become unavailable. You can select either **`Low`**, **`Normal`**, or **`High`** from the list. If there is only one router on the network, use **`Normal`**.

## Router Advertisement Subnets

This section allows you to define 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`*.)

## 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. Enter up to two IP addresses for DNS Servers, or leave blank to use the system default DNS servers or DNS forwarder if enabled. The Domain Search List operates identically to the DHCP option of the same name. If you check Use same settings as DHCPv6 server then 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's worth mentioning again here due to the differences in IPv6 addressing. Given the vast amount of space available inside even a /64, you may want to craft a range that restricts your hosts to use an easy to remember or recognize range. For example, Inside a /64 such as *`2001:db8:1:1::`*, you could have your DHCPv6 range be: *`2001:db8:1:1::d:0000`* to *`2001:db8:1:1::d:FFFF`*, using the *`d`* in the second to last section of the address as a sort of shorthand for "DHCP". That example range contains 2^16 (65,536) IPs, which is extremely large by today's IPv4 standards, but only a small portion of the whole /64.

# DHCPv6 Prefix Delegation

Prefix delegation, covered earlier in the section called “DHCP6 Prefix Delegation” and the section called “Track Interface”, allows you to automatically divide and allocate a block of IPv6 addresses to networks that will live behind other routers and firewall that reside downstream from pfSense (e.g. in

your LAN, DMZ, etc). Most users acting in a client capacity will not need this and will likely leave it blank.

You can use prefix delegation 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 IP 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 IPs specified here should be routed to this firewall by upstream routers. For example, to allocate /60 networks to downstream firewalls out of a given range, then you could specify *`2001:db8:1111:F000::`* to *`2001:db8:1111:FF00::`* with a Prefix Delegation Size of **`60`**. This would allocate 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 for your deployment.

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, you 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 you can start on *`2001:db8:1111:F500::`* but not *`2001:db8:1111:F550::`*.

# DHCPv6 Static Mappings

Static mappings on DHCPv6 work a little differently than those on 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 is where you enter the 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 you 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 you use a base image and clone workstations from there, they can all end up with the same DUID, and thus all end up pulling the same IPv6 address over DHCPv6.

You should clear the DUID from the registry before making your image to clone, by issuing the following command:

```
reg delete HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters /f
```

You can also run that command 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. If you need to provide DHCP service on a network segment without a DHCP server, you

use DHCP relay to forward those requests to a defined server on another segment. It is not possible to run both a DHCP server and a DHCP Relay at the same time. To enable the DHCP relay you must first disable the DHCP server on each interface.

Once the DHCP server is disabled, visit Services → DHCP Relay. As with the DHCP server, there is a tab for each interface. Click on the interface upon which you want run the DHCP relay, then check the box next to Enable DHCP relay on [name] interface, which will also let you set the other available options.

If you check Append circuit ID and agent ID to requests, the DHCP relay will append the 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, or may help distinguish where the requests originated.

The option to Proxy requests to DHCP server on WAN subnet does just what it says. If activated, it will pass DHCP requests from clients on this interface to the DHCP server which assigned the IP address to the WAN interface. Alternately, you may fill in the IP address of the DHCP server to which the requests should be proxied.

The DHCPv6 Relay function works identically to the DHCP Relay function for IPv4.

# DNS Forwarder

The DNS Forwarder in pfSense is a caching DNS resolver. It is enabled by default, and uses the DNS servers configured in System → General Setup, or those obtained from your ISP for dynamically configured WAN interfaces (DHCP, PPPoE, and PPTP). For static IP WAN connections, you must enter DNS servers in System → General Setup or during the setup wizard for the DNS forwarder to function. You can also use statically configured DNS servers 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.

In prior versions, pfSense initially tried the first configured DNS server when attempting to resolve a DNS name, and moved on to subsequently configured DNS servers if the first failed to resolve. This could cause long delays if one or more of the available DNS servers was unreachable. In pfSense 1.2.3 and later this behavior has been changed to query all DNS servers at once, and the only the first response received is used and cached. This results in much faster DNS service, and can help smooth over problems that stem from DNS servers which are intermittently slow or have high latency, especially in Multi-WAN environments. If you do not want this behavior, you can check Query DNS servers sequentially in the DNS forwarder options.

# 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

The DNS forwarder configuration is found under Services → DNS Forwarder.

## Enable DNS Forwarder

Checking this box turns on the DNS forwarder, or uncheck if you wish to disable this functionality.

## Register DHCP leases in DNS forwarder

If you want your internal machine names for DHCP clients to resolve in DNS, check this box. This only works for machines that specify a host name in their DHCP requests. The domain name from System → General Setup is used as the domain name on the hosts.

## Register DHCP static mappings in DNS forwarder

This works the same as the Register DHCP leases in DNS forwarder option, except that it registers the DHCP static mapping addresses.

## Resolve DHCP mappings first

When you have one IP address with multiple hostnames, doing a reverse lookup may give an unexpected result if you have one hostname in your 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 get you a result of *labserver01.example.com*, a test server's DHCP obtained IP, rather than a host override name of *testwww.example.com* you would otherwise get as the result.

## Query DNS servers sequentially

By default in pfSense 1.2.3 and later, pfSense queries all DNS servers simultaneously and uses the fastest result. This isn't always desriable, especially if you use OpenDNS and have site restrictions that could by bypassed by using a faster but less strict DNS server, or it could get results from a public DNS server over a private DNS server on the other end of a VPN. Checking this option goes back to the old behavior where it queries each DNS server in sequence from the top down, and waits for a timeout before moving on to the next DNS server in the list.

## Require domain

The Require Domain option, as its name implies, 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 IPs 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 DNS server of your choosing.

## 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 you might want to move the DNS Forwarder to another Listen Port, such as 5353 or 54, and then forward specific queries there via port forwards. The most common use case for that would be when running a DNS server package such as **tinydns**. When **tinydns** runs, it wants to bind to port 53 to answer queries, but it is only an authoritative name server, it won't handle recursive queries from clients. So the DNS Forwarder could handle connections via port forward on a specific interface or from a specific source, while **tinydns** could handle authoritative queries from outside.

## Interfaces

By default, the DNS Forwarder listens on every available interface and IPv4 and IPv6 address. The Interface control lets you limit 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

If this option is 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

In Advanced Options you can place any custom **dnsmasq** configuration parameters that are not configurable in the GUI. For example, to set a lower TTL for DNS records, you can enter `max-ttl=30`. Or you can craft a wildcard 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 [http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html].

## Host Overrides

The first section at the bottom of the DNS forwarder screen is where you can specify overrides for DNS host name resolution. Here you can configure a specific host name to resolve differently than it otherwise would via the DNS servers used by the DNS forwarder. This is useful for split DNS configurations (see the section called “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 IPs will be returned in the result. You can use this to supply both an IPv4 (A) and IPv6 (AAAA) result for a single hostname.

Figure 26.2, “DNS Override Example” illustrates a DNS override for an internal web server (example.com and www.example.com) as well as an example of blocking access to myspace.com and www.myspace.com.

**Figure 26.2. DNS Override Example**

| Host | Domain | IP | Description |
|---|---|---|---|
| | example.com | 192.168.1.100 | www override |
| | myspace.com | 127.0.0.1 | hack block |
| www | myspace.com | 127.0.0.1 | hack block |
| www | example.com | 192.168.1.100 | www override |

#### Note

It is not recommended to use strictly 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 is very easy to get around for those with more technical aptitude.

Host
: This field defines just the hostname of the DNS record (without the domain), e.g. `www`. It can be left blank if you are making 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.

Aliases — The aliases section allows you to define additional hostnames for this same IP address (much like CNAME records) if you wish to keep them in a single override entry.

## Domain Overrides

Domain overrides are found at the bottom of the DNS Forwarder screen. This allows you to specify a different 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's IP address ensures these records are resolved properly whether clients are using pfSense as a DNS server or the Windows Server itself.

In an Active Directory environment, your systems should always use your Windows DNS server as their primary DNS server so dynamic name registration functions properly. In environments with only one Windows DNS server, you should enable the DNS forwarder with an override for your Active Directory domain and use pfSense as the secondary DNS server for your 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 you alone to fix the problem if they can still check out their lolcats, MySpace, Facebook, et al in the mean time.

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 you usually want to resolve all DNS queries 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 you will need a static route for this to function over IPsec. See the section called “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 you like (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, if you have another entry for *example.com* to forward to *192.2.66.2*, but you want *lab.example.com* to forward on to your standard name servers, enter a *#* in this field. Third, it can be used to prevent non-local lookups by entering a *!*. If you have host override entries for *www.example.org* and *mail.example.org* but do not want any other lookups for hosts under *example.org* to be forwarded on to remote DNS servers, enter a *!* in this field.

Source IP — This field is optional, and mostly used by those who need to contact a DNS server on the other end of a VPN. Often only specific local IPs are able to traverse a VPN, this field lets you specify which IP address on the firewall is used to source the DNS.

Description — A text description used to identify or give more information about this entry.

# Dynamic DNS

The Dynamic DNS client in pfSense allows you to register the IP address of any WAN interface with a variety of dynamic DNS service providers. This is useful when you want to remotely access dynamic IP connections, most commonly used to connect to a VPN, web server, or mail server.

As pf pfSense 2.0, it now supports as many different dynamic DNS services as you desire, allows registration of OPT WAN IPs, and enables the registration of your real public IP in environments where pfSense receives a private IP for WAN and is NATed upstream.

# DynDNS and IPv6

As of this writing, there are very few DynDNS providers that offer IPv6 support. It should be possible to support providers easily in the future, but since there were so few providers we could find at the time to test against and implement any needed changes, it regrettably has to be marked as unsupported. The available choices are limited to HE.net when they host your domain's DNS, and RFC 2136 servers.

# Using Dynamic DNS

pfSense allows registration with seventeen different dynamic DNS providers as of version 2.1. You can see the available providers by clicking the Service type drop down box. You can find out more about those providers by searching for their name to find their web site. Most offer a basic level service at no cost, and some offer additional premium services at a cost. There is also a **`Custom`** option that lets you input in custom URL for an unsupported provider.

Once you decide on a provider, visit their website, register for an account and setup a hostname. The procedures for this vary for each provider, but they have instructions on their websites. After configuring your hostname with the provider, you then configure pfSense with those 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.

## Service Type

Select your dynamic DNS provider here.

## Interface to Monitor

Select the interface that has the IP you wish to keep updated, such as WAN, or an OPTx interface. By selecting a gateway group for the interface, the DynDNS entry can switch between WANs to allow for inbound Multi-WAN failover of services on this hostname.

## Hostname

Enter the hostname you created with your dynamic DNS provider. This should be the complete fully qualified domain name, such as *`myhost.example.com`*.

## MX

An MX (Mail Exchanger) record is how Internet mail servers know where to deliver mail for your domain. Some dynamic DNS providers will let you configure this via your dynamic DNS client. If yours does, enter the host name of the mail server that will receive Internet email for your dynamic DNS domain.

## Wildcards

Enabling wildcard DNS on your dynamic DNS name means all host name queries will resolve to the IP address of your dynamic DNS host name. For example, if your 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.

## Username and Password

This is where you enter the username and password for your dynamic DNS provider.

## 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, so the differences will be outlined for these in this section.

### Namecheap

When setting up DynDNS for your Namecheap domain, you receive an authentication token. This goes in the Password field, and the Username field is left blank.

### HE.net Tunnelbroker

The HE.net Tunnelbroker choice allows you to update your IPv6 tunnel endpoint IP when your WAN IP changes. The Hostname in this case is your Tunnel ID from HE.net.

### Route 53

When using an Amazon **`Route 53`** type, the Username is your Access Key ID provided by Amazon. You also must fill in the Zone ID you received when creating your domain in Route 53, and you can fill in the TTL for the DNS record also.

### Custom

The **`Custom`** DynDNS type lets you enter options that allow for updating otherwise unsupported services. In addition to the usual options, you can also select the Interface to send update from, which is almost always the same as the Interface, but can be changed as needed. When using the custom DynDNS type, the username and password fields are sent using HTTP basic authentication.

The Update URL would be the URL given by your DynDNS provider for use in updates. If the IP address must appear in the URL, enter it as *`%IP%`* and the real value will be substituted as needed.

The Result Match field lets you define expected output from the DynDNS 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. To disable checking of results, leave this box empty.

# RFC 2136 Dynamic DNS updates

The RFC 2136 dynamic DNS updates functionality allows you to register a hostname on any DNS server supporting RFC 2136 updates. This can be used to update hostnames on BIND and Windows Server DNS servers, amongst others. These entries are managed by navigating to Services → Dynamic DNS on the RFC 2136 tab.

This can run simultaneously with one of the previously discussed dynamic DNS service providers, and like those you may have as many entries as you need and they can be on any interface. RFC 2136 will update an A record, and an AAAA record if you have IPv6 configured on the interface to be monitored.

Configuring the server side of an RFC 2136 Dynamic DNS setup 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 here: http://doc.pfsense.org/index.php/RFC2136_Dynamic_DNS.

The client side is a bit different than the traditional Dynamic DNS types. There are a few more values required due to how the update request must be sent to the server. Most of the options are required.

| | |
|---|---|
| Enable | This checkbox controls whether or not the entry is active. If it is unchecked, the DNS update will not be performed for this entry. |
| Interface | The IP address on this interface will be given when performing the DNS update. |
| Hostname | The fully qualified domain name (FQDN) of the dynamic DNS entry to be updated. For example, *myhost.example.com*. |
| TTL | The Time To Live for the DNS entry. 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 might change. |
| Key Name | The name of the key as specified in the 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 | The 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 your DNS server administrator to find out the key type. Typically this is set to Host. |
| Key | This field contains the actual text of the key, such as */0/4bxF9A08n/zke/vANyQ==*. This value would be given to your by your DNS server or administrator. |
| Server | The IP address of the DNS server to which the update should be sent. |
| Protocol | When unchecked, the DNS update is sent over UDP, when checked it uses TCP for the update. |
| 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 Interface then a check is done to determine what the actual public IP address is, and then that IP address is used for the DNS update. |
| Description | A free-text description of the entry for your reference. |

As with the other DynDNS types, RFC 2136 updates are performed only when an IP change has been detected, or once every 25 days.

# SNMP

The Simple Network Management Protocol [http://en.wikipedia.org/wiki/Snmp] (SNMP) daemon will allow you to remotely monitor some pfSense system parameters. Depending on the options chosen, you can monitor 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.[1] In addition to the 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 you have the MIB II module loaded. The SNMP service can be configured by browsing to Services → SNMP.

The easiest way to see what data is available would be to run **snmpwalk** against the pfSense system from another host with **net-snmp** or an equivalent 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.

[1] http://people.freebsd.org/~harti/bsnmp/

# SNMP and IPv6

The **bsnmpd** implementation 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 all UDP, and SNMP clients default to using UDP port 161. This setting will cause the daemon to listen on a different port, and your SNMP client or polling agent should be changed to match. |
| System location | This text field specifies what string will be returned when the system's location is queried via SNMP. You may follow whatever convention is needed for your organization. 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 | The system contact is also a text field that can be set however your needs require. It could 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 you should consider changing it to something else 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 should be forwarded. |
| Trap server port | By default, SNMP traps are set on UDP port 162. If your 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

The loadable modules available here allow the SNMP daemon to understand and respond to queries for more system information. Each module loaded will consume additional resources. As such, ensure that only the modules that will actually be used 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, let you query 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 gives access to a wealth of information about pf. The MIB tree covers aspects of the ruleset, states, interfaces, tables, and ALTQ queues. |
| Host Resources | This module covers 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 you check this option, then MibII will be checked automatically. |
| UCD | This module provides a wealth of various system information knows as the ucdavis MIB, or UCD-SNMP-MIB. It provides information about memory usage, disk usage, running programs, and more. |
| Regex | The Regex module is mostly 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 will make the SNMP daemon listen on the chosen interface or virtual IP only. All interfaces with IP addresses, CARP VIPs, and IP Alias VIPs will be displayed in the drop-down list. This eases communications over VPN tunnels, as it eliminates the need for the previously mentioned static route, but it also helps provide some extra security by reducing the service's exposure on 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 and not the IP to which the query was sent.

# UPnP & NAT-PMP

Universal Plug and Play [http://en.wikipedia.org/wiki/Upnp] (UPnP) and NAT Port Mapping Protocol [http://en.wikipedia.org/wiki/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 creating their own NAT port forwards and associated firewall rules. The UPnP and NAT-PMP services on pfSense, found at Services → UPnP & NAT-PMP, will enable client PCs and other devices such as game consoles to automatically allow required traffic to reach them. There are many popular programs and systems which support UPnP, such as Skype, uTorrent, mIRC, IM clients, PlayStation 3, and XBox 360. 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. You may need to allow access to these services with firewall rules, especially if you have removed the default LAN-to-any rule, 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 was not fully IPv6 compatible, so if you are attempting to use it from an IPv6 enabled host, ensure the program is set to only expect incoming connections over IPv4.

## Security Concerns

The UPnP and NAT-PMP services are a classic example of the "Security vs. Convenience" trade-off. By their very nature, these services are insecure. Any program on the network could 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.

There are access controls present in the UPnP service configuration, which will help lock down who and what is allowed to make alterations. Over and above the built-in access controls, you can further control access 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

The UPnP and NAT-PMP services are configured by browsing to Services → UPnP & NAT-PMP. Enable the service by checking the Enable UPnP & NAT-PMP box and then you can selectively enable Allow UPnP Port Mapping, Allow NAT-PMP Port Mapping, or both. When you are finished making any needed changes, which are described in the remainder of this section, click Save. The UPnP and/or NAT-PMP service will then be started automatically.

## Interfaces

This setting lets you to pick the interfaces upon which UPnP is allowed to listen. More than one interface may be chosen by holding down **Ctrl** while clicking the additional interfaces. Deselecting an interface works the same way, hold **Ctrl** while clicking to remove the selection. If an interface is bridged to another, UPnP should only be selected on the interface that has the IP address, not a bridge member without an IP address. For example, if you have OPT1 bridged to LAN, only enable UPnP on LAN if LAN has the IP address for the bridge configured on it. If you have the bridge interface assigned and have the IP address configured there, then UPnP would only be enabled on the assigned bridge interface.

## Maximum Speeds

Starting with pfSense version 1.2.3, you may now set maximum download and upload speeds for ports opened by UPnP. These speeds are set in Kilobits per second, so to limit a download to 1.5Mbit/s, you would enter **1536** into the Maximum Download Speed field.

## Override WAN address

By default, the UPnP service will configure port forwards and firewall rules to the WAN address. This setting will let you enter an alternate IP address, such as a secondary WAN address or a shared CARP address.

## Traffic Shaping Queue

By default, rules created by UPnP will not assign traffic into a shaper queue. By entering the name of a queue into this field, traffic that passes due to a UPnP-created rule will fall into this queue. Choose the queue wisely, as any UPnP enabled device or program will use this queue. It could be Bittorrent, or it could be a game console, so choose a queue that has a priority that fits best with the traffic you expect to be most common.

## Log Packets

When this box is checked, the port forwards generated by UPnP 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.

## Default Deny

If the By default deny access to UPnP option is enabled, then 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.

## UPnP User Permissions

There are four fields for specifying user-defined access rules. If the default-deny option is chosen, you must set rules to allow access. Rules are formulated using the following format:

```
<[allow|deny]> <[external port|port range]> <[internal IP|IP/CIDR]>
<[internal port|port range]>
```

### UPnP User Permission Example 1

Deny access to port 80 forwarding from everything on the LAN, 192.168.1.1, with a /24 subnet.

```
deny 80 192.168.1.1/24 80
```

### UPnP User Permission Example 2

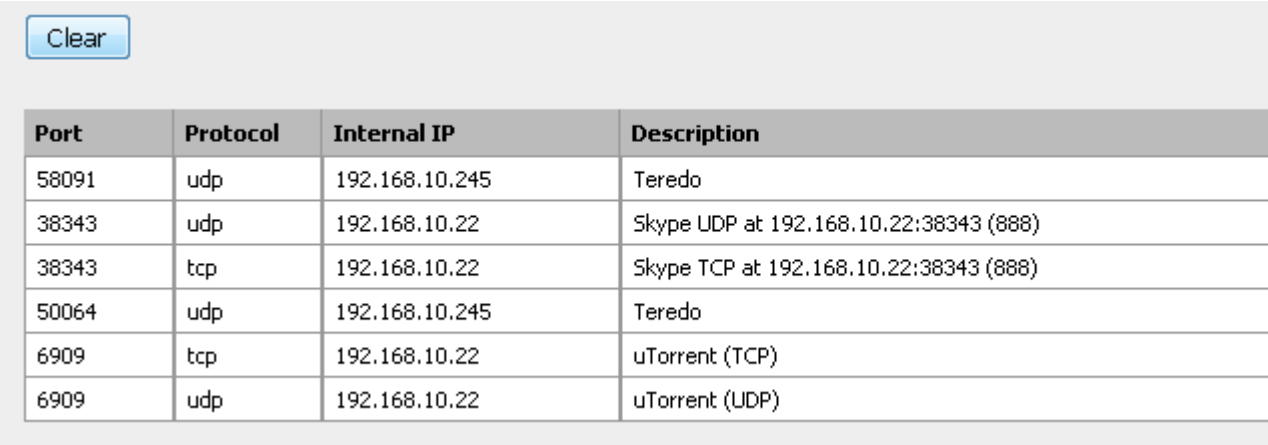
Allow 192.168.1.10 to forward any unprivileged port.

```
allow 1024-65535 192.168.1.10 1024-65535
```

# Status

The status of the UPnP service itself may be viewed at Status → Services. This will show if the service is running or stopped, and allow you to stop, start or restart the service. This should all be handled automatically, but may be controlled manually if needed. A list of currently forwarded ports and clients like that in Figure 26.3, “UPnP & NAT-PMP status screen showing client PCs with forwarded ports” may be viewed under Status → UPnP & NAT-PMP.

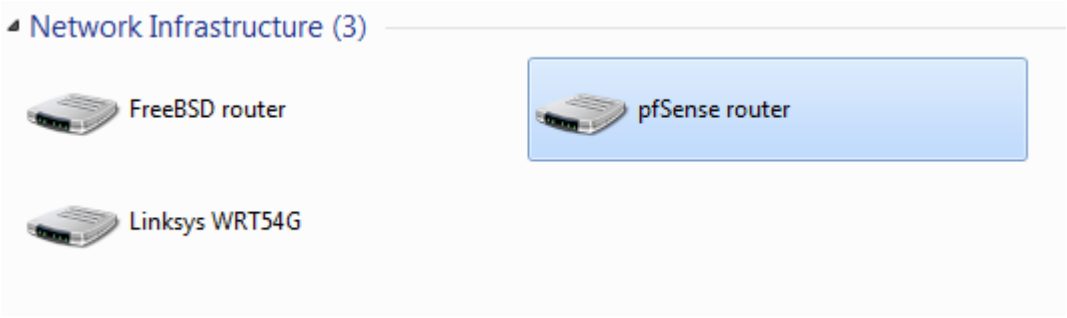
**Figure 26.3. UPnP & NAT-PMP status screen showing client PCs with forwarded ports**




When the service is running it should also show up when you browse the network using a UPnP-aware Operating System like Windows 7, 8 or Vista, as shown by Figure 26.4, “pfSense system as seen by

Windows 7 when browsing the Network". You can right click on the router's icon and then click View device webpage to open up the WebGUI in your default browser. If you right click on the router and click Properties, it will also show the pfSense version and IP address of the router.

**Figure 26.4. pfSense system as seen by Windows 7 when browsing the Network**




# Troubleshooting

Most issues with UPnP tend to involve bridging. In this case it is important that you have specific firewall rules to allow UPnP on UDP port 1900. Since it is multicast traffic, the destination should be the broadcast address for the subnet, or in some cases making it **any** will be necessary. Consult your 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 the section called "Static Port" for more details.

# NTPD

The NTP [http://www.ntp.org/] service is a Network Time Protocol [http://en.wikipedia.org/wiki/Network_Time_Protocol] (NTP) daemon which will listen for requests from clients and allow them to synchronize their clock with that of the pfSense system. By running a local NTP server and using it for your clients, it reduces the load on the lower-stratum servers and can ensure that your systems can always reach a time server. Before delegating this task to your pfSense system, it is a good practice to ensure that it has an accurate clock and keeps time reasonably.

In pfSense 2.0.1 and before, the OpenNTPD package was used for this task. As of pfSense 2.0.2, it was replaced with the NTP Project implementation which is the standard daemon used on FreeBSD.

There is not much to configuring the NTP server, available at Services → NTP. On this screen you can pick which Interfaces or Virtual IPs it should listen upon, and click Save. More than one interface may be chosen by holding down **Ctrl** while clicking the additional interfaces. Deselecting an interface works the same way, hold **Ctrl** while clicking to remove the selection. The service will be started immediately, however there will be a several minute delay before it will service NTP requests, as the service ensures its time is accurate before answering requests.

## Note

The NTP Project daemon binds to all interfaces by default, because it must do that to properly receive replies. You may choose to minimize this binding by selecting at least one interface, but that interface will also be used to source the NTP queries sent out to remote servers, not just to serve clients. Deselecting all interfaces in the config is now the equivalent of selecting them all.

# NTP and IPv6

The NTP Project daemon fully supports IPv6 as a client and a server.

# Logging

NTP logs are kept under Status → System Logs, on the NTP tab. NTP has very little logging, unless there is a problem the service will never generate any log entries.

# Serial GPS

If your system has an available serial port, an additional section on the page appears so that you may configure a Serial GPS to provide a reference clock for the firewall. The GPS must provide NMEA format output.

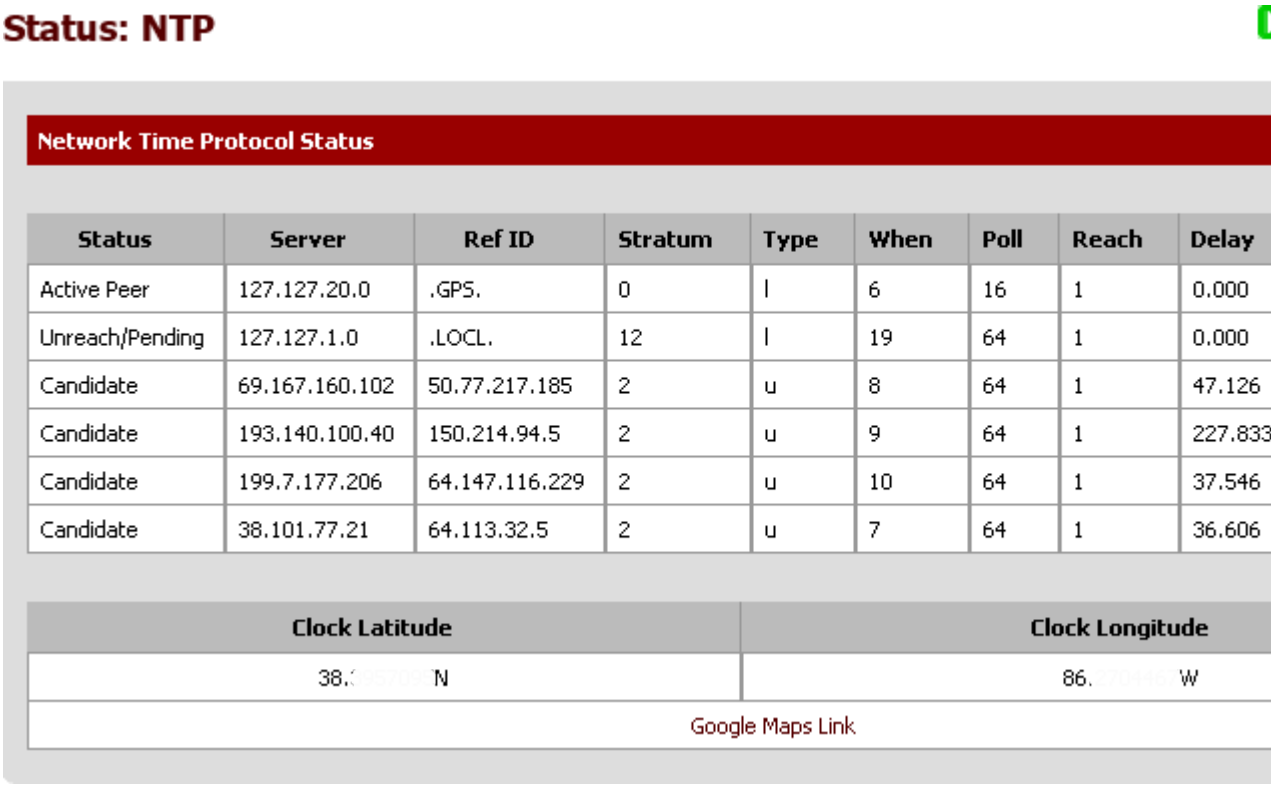
All serial ports detected on the system are listed in the drop-down box, so be sure to pick only the serial port to which the GPS is attached.

It is best to configure at least 2 servers under System → General 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.

# Status

New with the switch to the NTP Project daemon is the ability to have a status page indicating the clock's synchronization status with its peers. This status page can be found at Status → NTP, and looks like Figure 26.5, “NTP Daemon Status with GPS output”.

**Figure 26.5. NTP Daemon Status with GPS output**




The status screen contains one line for every peer, and lists the peer's 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 tells you which peer is currently the active peer for time sync, and which servers are potential candidates to be peers, or which ones have been rejected and why.

If you have a serial GPS connected, 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 your 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 works, 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 [http://en.wikipedia.org/wiki/Wake_on_lan] (WOL) page at Services → Wake on LAN can be used to wake up computers from a powered-off state by sending special "Magic Packets". The NIC in the computer that is to be woken up must support WOL and has to be configured properly. Typically there is a BIOS setting to enable WOL, and non-integrated adapters likely need 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 powered off, and conserve power. Should a service be 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 machine can be awoken, and may then be accessed once it has booted.

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 a couple of vendor-specific WOL extensions that provide some extra security, but nothing universally supported.

## Wake Up a Single Machine

To wake up a single machine, choose the Interface through which it can be reached, and enter the system's MAC address in the format of *xx:xx:xx:xx:xx:xx*. When you click Send, pfSense will transmit a WOL Magic Packet out the chosen interface, and if everything went as planned, the system should 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 later convenience, click the by the list of stored MAC addresses, and you will see a blank edit screen. Pick the Interface through which it can be reached, and enter the system's MAC address in the format of *xx:xx:xx:xx:xx:xx*. A description may also be entered for later reference, for example "Pat's PC" or "Sue's Server". Click Save when finished and you will be returned to the main WOL page and your new entry should be visible in the list at the bottom of the page.

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, click its MAC address in the list of stored systems. You will be taken back to the WOL page, and the Magic Packet will be sent automatically.

## Wake All Stored Machines

On the WOL page, there is a button which can be used to send a WOL Magic Packet to all stored systems. Click the button and the requests will be sent, with no other intervention required.

## Wake from DHCP Leases View

To send a WOL Magic Packet from the DHCP Leases view at Diagnostics → DHCP leases, click its MAC address in the list of leases, which should be highlighted as a link. The WOL link will only be active for systems whose status is shown as "offline". You will be taken back to the WOL page, and the Magic Packet will be sent automatically.

## Save from DHCP Leases View

You can copy a MAC address to a new WOL mapping entry while viewing the DHCP leases at Diagnostics → DHCP leases. Click the button at the end of line, and you will be taken to the WOL entry edit screen with that system's information pre-filled in the form. Add a description, and then click Save.

# PPPoE Server

pfSense can act as a PPPoE server and accept/authenticate connections from PPPoE clients on a local interface, acting as an access concentrator. This can be used to force users to authenticate before gaining network access, or otherwise control their login behavior. This is found under Services → PPPoE Server. You will find that this configuration is very similar to the PPTP VPN server (Chapter 19, *PPTP VPN*).

Starting with pfSense 2.0, you can have multiple PPPoE servers on different interfaces. To begin setting up a PPPoE server, click and you will be taken to a page to edit the values for a PPPoE server instance.

To turn on this instance, you must first select Enable PPPoE server. Then choose which Interface on which to offer this service. Set the Subnet Mask which should be assigned to PPPoE clients and the Number of PPPoE Users to allow. Now enter the Server Address which is the IP address which the pfSense system will send to the PPPoE clients to use as their gateway. This IP address should NOT be an IP address currently in use on the firewall. Enter an IP address in the Remote Address Range box and that will be used together with the Subnet Mask set earlier to define the network used by the PPPoE clients.

You may optionally enter a Description for this server instance if desired. The DNS Server fields can be used to send specific DNS servers to the PPPoE clients, otherwise the firewall's IP will be sent to the client for DNS if you have the DNS forwarder enabled. If the DNS forwarder is disabled, then the DNS servers configured on the firewall will be sent instead.

The remaining options are for authentication via RADIUS. If you wish to pass the authentication requests on to a RADIUS server, fill in the information on the lower half of the screen. If you would instead prefer to use local authentication, then click the next to Users to add local users. Click once for each user you would like to add, and then fill in the username, password, and an optional IP address.

See the section called “RADIUS Authentication with Windows Server” for information on setting up RADIUS on a Windows server, but you may use whichever RADIUS server you prefer.

# IGMP Proxy

In pfSense 2.0, an IGMP Proxy service was integrated. It can be Services → IGMP Proxy. From there, you should add at least one upstream interface and one downstream interface. Interfaces are added by clicking .

After clicking the Edit screen appears to edit an interface entry. First, select the interface to be used by the proxy. Next, you may optionally enter a Description for this interface instance.

Next, choose a Type. This can either be an **Upstream Interface** or a **Downstream Interface**. The upstream network interface is the outgoing interface which is responsible for communicating to available multicast data sources. There can only be one upstream interface. Downstream network interfaces are 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.

The Threshold parameter defines 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.

Lastly, you can add CIDR-masked Network entries to control what subnets are allowed to have their multicast data proxied.

# Chapter 27. System Monitoring

As important as the services provided by pfSense is the data and information that pfSense lets you see. 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).

# System Logs

pfSense logs quite a bit of data by default, but does so in a manner that will not overflow the storage on the router. The logs are found under Status → System Logs in the WebGUI, and under `/var/log/` on the filesystem. Some components such as DHCP and IPsec, among others, generate enough logs that they have their own logging tabs to reduce the clutter in the main system log and ease troubleshooting for these individual services. To view these other logs, click the tab for the subsystem you want to view. Certain things, like the System, and VPN tabs, have sub-tabs with more related options grouped together.

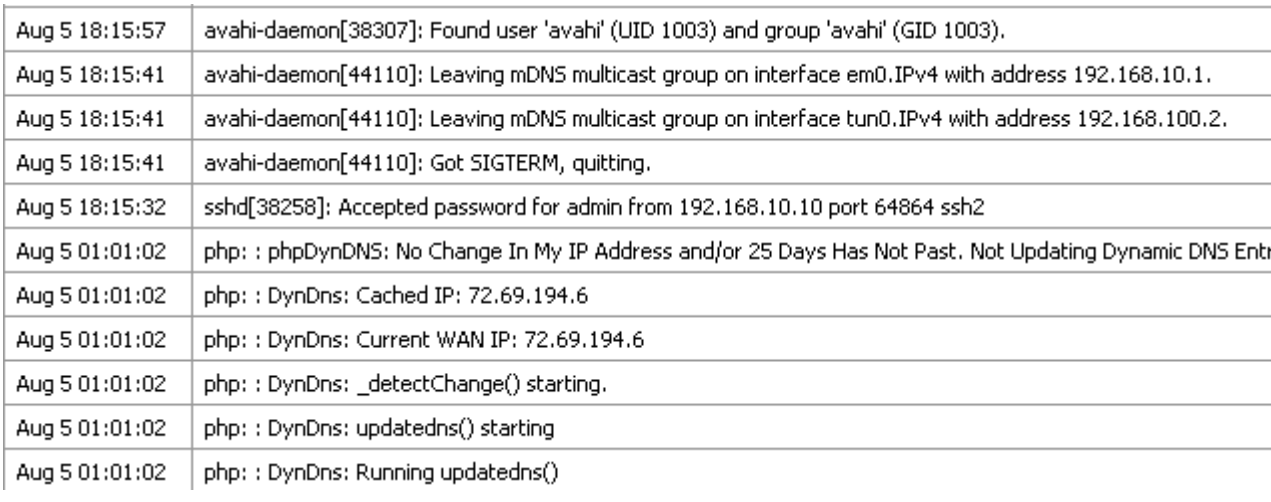
pfSense logs are contained in a binary circular log or *clog* format. These files are a fixed size, and never grow. As a consequence of this, the log will only hold a certain amount of entries, and the old entries are continually pushed out of the log as new ones are added. If this is an issue for you or your organization, you may adjust the log settings to copy these entries to another server with syslog where they may be permanently retained or rotated with less frequency. See the section called “Remote Logging with Syslog” later in this section for information about syslog. Starting with pfSense 2.0.2, the log files are retained at bootup on a full install, and their sizes have been slightly increased from previous versions.

## 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 some services and packages which do not have their logs redirected to other tabs/log files.

As you can see by the example entries in Figure 27.1, “Example System Log Entries”, there are log entries from the SSH daemon, the avahi package, and the dynamic DNS client. 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/log file. Also note in this example that the logs are configured to appear in reverse order, and the newest entries appear at the top of the list. See the next section to find out how to configure the logs for reverse order.

**Figure 27.1. Example System Log Entries**



| | |
|---|---|
| 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 Entr |
| 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() |

# Changing Log Settings

Log settings may be adjusted by going to Status → System Logs and using the Settings tab. Here you will find several options to choose from that control how logs are displayed.

The first option, Show log entries in reverse order, controls the order in which logs are displayed on the various logging tabs. With this option checked, the newest entries will be at the top of the log output. When this option is unchecked, the oldest entries will be at the top. Certain people find both of these methods useful and easier to follow, so you can pick whichever setting you prefer.

The next setting, GUI Log Entries to Display, only controls how many log lines are displayed on each tab. The actual logs may contain more data, so this can be adjusted up or down a bit if needed to show more log data on each tab.

Normally, every packet blocked by the firewall's default deny rule is logged. If you do not want to see these log entries, uncheck the Log packets blocked by the default rule option. Similarly, there are options to control Log packets blocked by 'Block Bogon Networks' rules and Log packets blocked by 'Block Private Networks' rules to toggle whether or not their respective rules generate log messages.

By default, the WebGUI web server process, lighttpd, will log errors to the main system log. This is OK in most cases but in certain environments the logs can be very noisy unnecessarily, and the Log errors from the web server process option will disable the logging from that process.

The Show raw filter logs option controls the output of the Firewall logs tab. When checked, the output will not be interpreted by the log parser, and will instead be displayed in its raw format. Sometimes this can aid in troubleshooting, or if you need support the raw log will give a technician more information than is normally seen in the default firewall log output. The raw logs are harder to read and interpret than the parsed logs, so this is typically left unchecked most of the time.

The Filter descriptions drop-down governs how firewall log entry descriptions are displayed. By default, with **`Don't load descriptions`**, they are not shown, but you can click the action icon to see the associated rule description. **`Display as column`** will put the rule description into an additional column. **`Display as a second row`** will put the description in its own row below the log entry, in a collapsible way.

If you would also like to disable local logging, you can check Disable writing log files to the local disk but this is not generally recommended.

Click Save when you are done making changes. The remaining options on this screen are discussed in the following section.

# Remote Logging with Syslog

The other options under Status → System Logs on the Settings tab are for using syslog to copy log entries to a remote server. Because the logs kept by pfSense on the router itself are of a finite size, copying these entries to a syslog server can help with troubleshooting and long-term monitoring. The logs on the router are of a fixed size, and they are cleared on reboot on NanoBSD, so having a remote copy can also help diagnose events that occur just before a router restarts or after they would have otherwise scrolled off the log.

Some corporate or legislative policies dictate how long logs must be kept for firewalls and similar devices. If your organization requires long-term log retention, you will need to configure a syslog server to receive and retain these logs.

To start logging remotely, check Send log messages to remote syslog server, and fill in an IPv4 address, IPv6 address, or hostname for up to three syslog servers in the Remote Syslog Servers section.

The syslog server is typically a server that is directly reachable from your pfSense system on a local interface. Logging can also be sent to a server across a VPN, but may need some extra configuration

(see the section called “pfSense-initiated Traffic and IPsec”) You should not send syslog data directly across your WAN connection, as it is plain text and could contain sensitive information.

Check the boxes for the log entries you would like copied to the syslog server. You can choose to remotely log Everything, or individually select any of System events, Firewall events, DHCP service events, Portal auth events, VPN events, Gateway monitoring events, Server load balancer events, or Wireless events.

Be sure to click Save when you are finished making changes.

If you do not have a syslog server, it is fairly easy to set one up. See the section called “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 FreeBSD requires only a couple steps. In these examples, replace *192.168.1.1* with the IP address of your firewall, replace *exco-rtr* with the hostname of your firewall, and replace *exco-rtr.example.com* with the full hostname and domain of your firewall. I use *192.168.1.1* in these examples because it is recommended to do this with the *internal* address of your router, not a WAN type interface.

First, you will likely need an entry in `/etc/hosts` that contains the address and name of your firewall, like so:

```
192.168.1.1            exco-rtr    exco-rtr.example.com
```

Then you need to adjust **syslogd**'s startup flags to accept syslog messages from the firewall. Edit `/etc/rc.conf` and add this line if it doesn't exist, or add this option to the existing line for the setting:

```
syslogd_flags=" -a 192.168.1.1 "
```

Lastly, you'll need to add some lines to `/etc/syslog.conf` that will 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, and then set a host filter for your firewall (use its short name as entered in `/etc/hosts`). If you are familiar with syslog, you can look at `/etc/syslog.conf` on the pfSense router and also filter the logs for various services into separate log files on the syslog server.

After these changes you will need to restart **syslogd**. On FreeBSD this is just one simple command:

```
# /etc/rc.d/syslogd restart
```

You should now be able to look at the log file on the syslog server and see it populating with log entries as activity happens on the firewall.

# Dashboard

After finishing the Setup Wizard, you will end up at the main page of the firewall, which is the Dashboard. The Dashboard page, introduced in pfSense 2.0, greatly improves the quantity and quality

of information that can be seen at a glance on the firewall's main page. The same system information is shown as on previous versions of pfSense, along with much more. Many other types of information are available in separate widgets. These widgets can be added or removed, and dragged around to the positions desired by the user.

# Managing Widgets

Each widget follows some basic conventions for controlling its position, size, settings, etc. The mechanics of these operations are covered here, before we move on to the individual widgets and their capabilities.

## Adding and Removing Widgets

To start adding widgets, click the button at the top of the Dashboard and the list of widgets will be shown. Click on the name of a widget to add it to the Dashboard, and then it will appear in one of the columns. Once the widget has been added, click Save Settings.

**Figure 27.2. Widget Title Bar**


To close and remove a widget from the Dashboard, click the button in its title bar, as seen in Figure 27.2, “Widget Title Bar”, then click Save Settings.

## Rearranging Widgets

Widgets can be rearranged and moved between columns. To move a widget, click and drag its title bar (Figure 27.2, “Widget Title Bar”), move the mouse to the desired position, and then release. As the widget is moved it will "snap" into its new position, so you can see its new location before releasing the mouse button. After positioning a widget, click Save Settings.

## Minimizing Widgets

To minimize a widget so it only shows up as its title bar, hiding the content, click the button in its title bar, as seen in Figure 27.2, “Widget Title Bar”. To restore the widget to its normal display, click the button. After changing the widget's view, click Save Settings.

## Changing Widget Settings

Some widgets have customizable settings that control how their data is displayed or updated. If a widget has settings, the button will show up in its title bar, as seen in Figure 27.2, “Widget Title Bar”. Click that button and the widget's settings will appear. Once you have adjusted the settings, click Save inside of the widget.

# Available Widgets

Each widget contains a specific set of data, type of information, graph, etc. Each of the currently available widgets will be covered in this section, along with their settings (if any). These are listed in alphabetical order.

## Captive Portal Status

This widget shows the current list of online captive portal users, including their IP address, MAC address, and username.

## CARP Status

The CARP Status widget displays a list of all CARP type Virtual IP addresses, along with their status as either MASTER or BACKUP.

## Gateways

The Gateways widget lists all of the system's gateways along with their current status. The status information consists of the gateway's 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. Packages may be updated from this widget by clicking the  button at the end of a package's row.

## 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 each interface's name, IPv4 address, IPv6 address, the interface's link status (up or down), as well as the link speed.

## IPsec

The IPsec widget has two tabs for its two separate views. 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.

## 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-udpating 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.

## 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, for SSL/TLS based servers it will show a list of all connected clients. For static key clients and servers, it will show an up/down status. In each case it displays the IP address of the connecting client with the name and time of the connection.

## Picture

The picture wizard displays a picture of your choice inside of a widget. 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's tool bar, click Browse to find the picture on your computer, and click Upload to upload the picture. As mentioned in the note on the widget, the best pictures are within the dimensions of 350 pixels wide by 350 pixels high.

## RSS

The RSS (RDF Site Summary, or as it's often called, Really Simple Syndication) widget will display an RSS feed of your choosing. 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.

## 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 your hard drive, this will show a brief status of the drive's integrity as reported by S.M.A.R.T.

## System Information

This widget is the main widget, displaying a multitude of information about the running system. The information displayed includes:

| | |
|---|---|
| Name | The configured 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. Clicking the FreeBSD version will show the complete details of the running kernel version.<br><br>Under those items is the result of an automatic update check for a more recent version of pfSense. This automatic update check can be disabled in the firmware settings. |
| Platform | The platform indicates which variation of pfSense is running. A full install will show pfSense, an embedded install shows nanobsd, and if running from the LiveCD or memstick it will show cdrom. |
| 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 CPU type displayed here is the version string for the processor, such as "Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz". If powerd is active and the CPU frequency has been lowered, then the current frequency is shown along size the maximum frequency. |

| | |
|---|---|
| Hardware crypto | If a known hardware cryptographic accelerator has been detected, it will be displayed here. There may be other cards supported by FreeBSD that work correctly but are not detected by this widget because we did not have access to the specific strings needed to detect them. If you have such a card, contacting us with information about it would be appreciated. |
| 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, should the time zone you are viewing the firewall from be different from where it resides, you can tell the firewall's local time from this line. |
| 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 the number 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 memory buffers in use, and the maximum the system has available. These memory buffers are used for network operations, among other tasks. If the number is close to maximum or at the maximum, you should increase the number of available mbufs as described in the section called “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 your platform. So on slower platforms such as ALIX this is likely to read a bit 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 hard drive or storage media. |

## Traffic Graphs

The Traffic Graphs widget gives you a live SVG graph for the traffic on each interface. Each graph can be individually expanded or minimized by clicking the individual graph title. The default refresh rate of the graphs is once every 10 seconds, but that may be adjusted in the widget's settings. 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.

# Interface Status

**Figure 27.3. Interface Status**

| DSL interface (pppoe0) | |
|---|---|
| Status | up |
| PPPoE | up Disconnect |
| Uptime | 290:31:15 |
| MAC address | 00:00:00:00:00:00 |
| IPv4 address | 50.127.71.251 |
| Subnet mask IPv4 | 255.255.255.255 |
| Gateway IPv4 | 74.42.148.105 |
| IPv6 Link Local | fe80::240:48ff:feb2:8216 |
| ISP DNS servers | 127.0.0.1<br>8.8.8.8<br>74.40.74.40<br>2001:4860:4860::8888<br>2001:4860:4860::8844 |
| In/out packets | 9298168/9298168 (4.49 GB/710.76 MB) |
| In/out packets (pass) | 9298168/6700957 (4.49 GB/710.76 MB) |
| In/out packets (block) | 4757/0 (626 KB/0 bytes) |
| In/out errors | 0/0 |
| Collisions | 0 |

| LAN interface (em0) | |
|---|---|
| Status | up |
| MAC address | 00:40:48:b2:82:16 |
| IPv4 address | 192.168.20.1 |
| Subnet mask IPv4 | 255.255.255.0 |
| IPv6 Link Local | fe80::240:48ff:feb2:8216 |
| IPv6 address | 2001:470: ::1:1 |
| Subnet mask IPv6 | 64 |
| Media | 1000baseT <full-duplex> |
| In/out packets | 47071331/48838220 (6.88 GB/43.15 GB) |
| In/out packets (pass) | 47071331/55361598 (6.88 GB/43.15 GB) |
| In/out packets (block) | 93627/7 (36.15 MB/340 bytes) |
| In/out errors | 0/0 |

The status of the network interfaces may be viewed at Status → Interfaces. In the first part of Figure 27.3, “Interface Status”, a PPPoE WAN connection has been made and the IP, DNS, etc has been obtained. You can also see the network interface's MAC address, media type, in/out packets, errors, and collisions. Dynamic connection types like PPPoE and PPTP have a Disconnect button when

connected and a Connect button when offline. Interfaces obtaining an IP 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, you can see the LAN connection. Since this is a normal interface with a static IP, only the usual set of items are shown.

If an interface's status says "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 you may also see errors and collisions due to a link speed or duplex mismatch. See the section called “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 27.4, “Services Status”. The status is usually listed as Running or Stopped. From this view, a running service may be restarted by clicking or stopped by clicking . A stopped service may be started by clicking . Normally, it is not necessary to control services in this manner, but occasionally there may be maintenance or troubleshooting reasons for doing so. If available, other shortcuts are shown to take you to a service's configuration (), detailed status page (), or logs ().

**Figure 27.4. Services Status**

| Service | Description | Status |
|---|---|---|
| bsnmpd | SNMP Service | Running |
| cron | The cron utility is used to manage commands on a schedule. | Running |
| dhcpd | DHCP Service | Running |
| dnsmasq | DNS Forwarder | Running |
| miniupnpd | UPnP Service | Running |
| ntpd | NTP clock sync | Running |
| openvpn | OpenVPN client: Cluster | Running |
| openvpn | OpenVPN client: NYI site to site | Running |
| Quagga OSPFd | Not available. | Running |
| Quagga Zebra | Not available. | Running |
| racoon | IPsec VPN | Running |
| radvd | Router Advertisement Daemon | Running |

# RRD Graphs

RRD Graphs are another useful set of data provided by pfSense. While the router is running it keeps track of various bits of data about how the system performs, and then stores this data in Round-Robin Database (RRD) files. Graphs of this data are available from Status → RRD Graphs. On that screen there are six tabs, each of which are covered in this section: System, Traffic, Packets, Quality, Queues, QueueDrops, VPN, Custom, and Settings.

Each graph is available in several times spans, and each of these is averaged over a different period of time based on how much time is being covered in a given graph. Also on each graph will be a legend and a summarization of the data being shown (minimums, averages, maximums, current values, etc.).

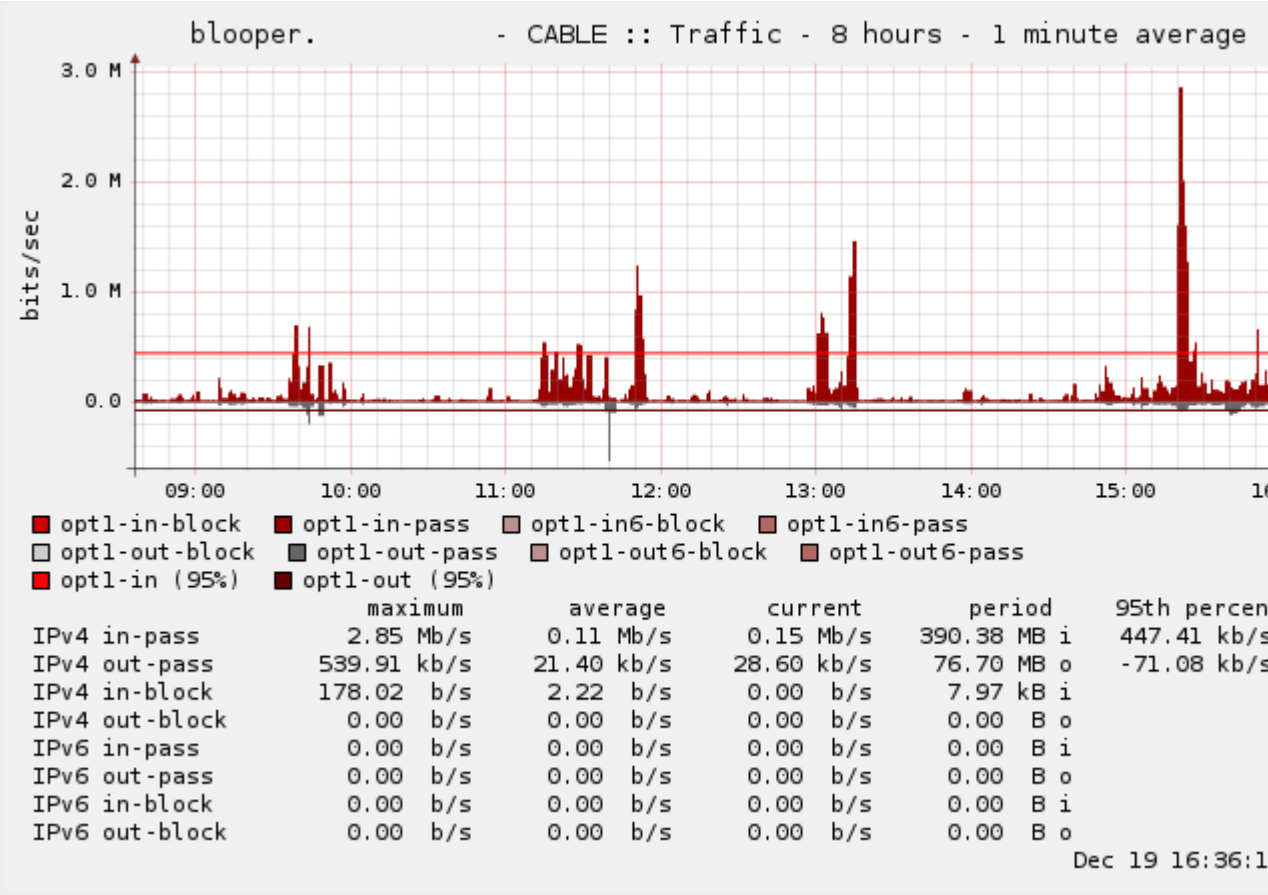
Graphs are available in an 8 hour range with a 1 minute average, a 1 day range with a 5 minute average, a 1 week range with a 1 hour average, a 1 month range with a 1 hour average, a 3 month range with a 1 day average, a 1 year range with a 1 day average, and a 4 year range with a 1 day average.

Many graphs can be viewed in Inverse style or Absolute style. With Inverse style, the graph is split down the middle horizontally and incoming traffic is shown going up from the center, and outgoing traffic is shown going down from the center. With Absolute style, the values are superimposed.

The graph's Period defaults to Absolute Timespans which shows the expected values for time. Current Period will start the graph at the boundary for which it is being drawn, for example the one hour graph starts at the top of the hour. Previous period works similarly, but for the preceding time span.

In Figure 27.5, “WAN Traffic Graph”, you can see that it is an 8 hour inverse graph of traffic on the CABLE interface, which has had a maximum use of 2.85Mbit/s during a 1 minute period.

**Figure 27.5. WAN Traffic Graph**




# System Graphs

The graphs under the System tab show a general overview of the system utilization, including CPU usage, total throughput, and firewall states.

## Processor Graph

The processor graph shows CPU usage for user and system processes, interrupts, and the number of running processes.

## Throughput Graph

The throughput graph shows the incoming and outgoing traffic totalled up for all interfaces.

### States Graph

The states graph is a bit more complex. It shows the number of system states but also breaks down the value in several ways. It shows the filter states from firewall rules, NAT states from NAT rules, the count of unique active source and destination IP addresses, and the number of state changes per second.

### Memory Graph

The Memory graph shows how the system is using RAM. There are lines on the graph showing the amount of free memory (not used at all), active memory (in use), inactive memory (was in use, could be reallocated), memory used for caching, and wired memory (typically kernel memory). The OS will attempt to use RAM as much as it can for caching rather than letting it sit idle, so don't be surprised if the amount of free RAM is lower than expected.

## Traffic Graphs

Traffic graphs will show the amount of bandwidth used on each available interface in bits per second notation, and there is also an Allgraphs choice which will show all of the traffic graphs on a single page.

## Packet Graphs

The packet graphs work much like the traffic graphs, except instead of reporting based on bandwidth used, it reports the number of packets per second (pps) passed.

## Quality Graphs

The quality graph tracks the quality of WAN or WAN-like interface (those with a gateway specified, or using DHCP, PPPoE, etc.). Shown on these graphs are the response time from the gateway in milliseconds, as well as a percentage of lost packets. Any loss on the graph indicates connectivity issues or times of excessive bandwidth use.

## Queue/Queuedrops Graphs

The queue graphs are a composite of each traffic shaper queue. Each individual queue is shown, represented by a unique color. You can view either the graph of all queues, or the graph representing the drops from all queues. The Queuedrops graphs show the number of drops on each queue.

## VPN Graphs

The VPN graphs tab shows the number of users logged in over VPN sessions at a given time.

## Custom Graphs

The custom graphs tab, as the name implies, lets you make a custom graph image from any of the various graph database files available on the other tabs. From the Graphs drop-down, select the graph database to use. Then select a Style and set a Start and End time. Clicking in the Start or End fields will produce a calendar pop-up that can be used to choose the timespan for the graph. Click Go and the graph will be shown.

## Settings

The RRD graphs can be customized to better suit your preferences. You can even turn them off if you prefer to use some external graphing solution instead. Click Save when finished making changes.

### Enable Graphing

Check the box to turn on graphing, or remove the check to disable graphing.

### Default Category

The Default Category option picks which tab will show up first when you click on Status → RRD Graphs.

### Default Style

The Default Style option picks which style of graphs to use by default, Inverse or Absolute.

### Default Period

Let you choose the default Period for graphs. The default value is Absolute Timespans which shows the expected values for time. Current Period will start the graph at the boundary for which it is being drawn, for example the one hour graph starts at the top of the hour. Previous period works similarly, but for the preceding time span.

### Reset RRD Data

This button will remove all RRD graph database files and start them over fresh. This can be necessary if a database file has become corrupt, or if the system changed architectures between i386 and amd64 which renders the data inaccessible.

# Firewall States

As discussed in the section called “Stateful Filtering”, pfSense is a stateful firewall and uses one state to track each connection to and from the system. These states may be viewed in several ways, either in the WebGUI or from the console.

# Viewing in the WebGUI

Viewing the states from the WebGUI can be done by visiting Diagnostics → States (Figure 27.6, “Example States”). Here you will see the protocol for each connection, its Source, Router, and Destination, and its connection state. When dealing with NAT entries, the three entries in the middle column represent the system which made the connection, the IP address and port pfSense is using for the NAT connection, and the remote system to which the connection has been made.

Individual states may be removed by clicking the at the end of their row.

**Figure 27.6. Example States**

| | | |
|---|---|---|
| tcp | 192.168.10.10:53650 -> 72.69.194.6:41047 -> 168.143.168.68:443 | FIN_WAIT_2:FIN_WA |
| udp | 224.0.0.251:5353 <- 192.168.10.17:5353 | NO_TRAFFIC:SINGLE |
| tcp | 207.45.186.18:80 <- 192.168.10.11:1289 | ESTABLISHED:ESTAB |
| tcp | 192.168.10.11:1289 -> 72.69.194.6:52740 -> 207.45.186.18:80 | ESTABLISHED:ESTAB |

### States Summary

The State Table Summary, accessible from Diagnostics → States Summary, provides some statistics about the state table and connections that are currently in it. The state table contents are analyzed and then it shows the IP, along with a total state count, and a breakdown 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 your environment, high values by any metric may be normal.

The following information categories are shown:

| | |
|---|---|
| Connections listed by source IP | Summarized by the source IP of the connection. This is useful for finding a potential source of attack, or a port scan or similar type probe/attack. |
| Connections listed by destination IP | Summarized by the destination IP of the connection. Useful for finding the target of an attack, or identifying servers. |
| Total per IP | Summarized by all connections to or from an IP. Useful for finding very active hosts using lots of ports, such as bittorrent clients. |
| By IP Pair | Summarizes states between two IPs involved in active connections. Useful for finding specific client/server pairs that have unusually high numbers of connections. |

### Note

The States Summary can take a long time to process and display, especially if you have a very large state table. In cases where the state table is extremely large, the page may not display properly, or it may fail with a memory error.

## Viewing with pftop

**pftop** is available from the GUI and the system console menu, and offers a live view of the state table along with the total amount of bandwidth consumed by each state. In the GUI, only the default view is available from under Diagnostics → pfTop. From the console, there are several ways to alter the view while watching pfTop. 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.

## Source Tracking States

When using Sticky connections (the section called “Sticky Connections”), the firewall also maintains a source tracking table that records which internal IPs map to which external gateway. These associations, by default, only exist so long as there are any active states from the internal IP. There is a configurable timeout for these source tracking entries to allow them to exist longer if needed (the section called “Sticky Connections”).

The source tracking associations are shown on Diagnostics → States on the Source Tracking tab. There you can see the Source to Destination mapping, and some other related data. As with states, these associations can be individually removed by clicking the at the end of their row.

## Reset State Table / Source Tracking Table

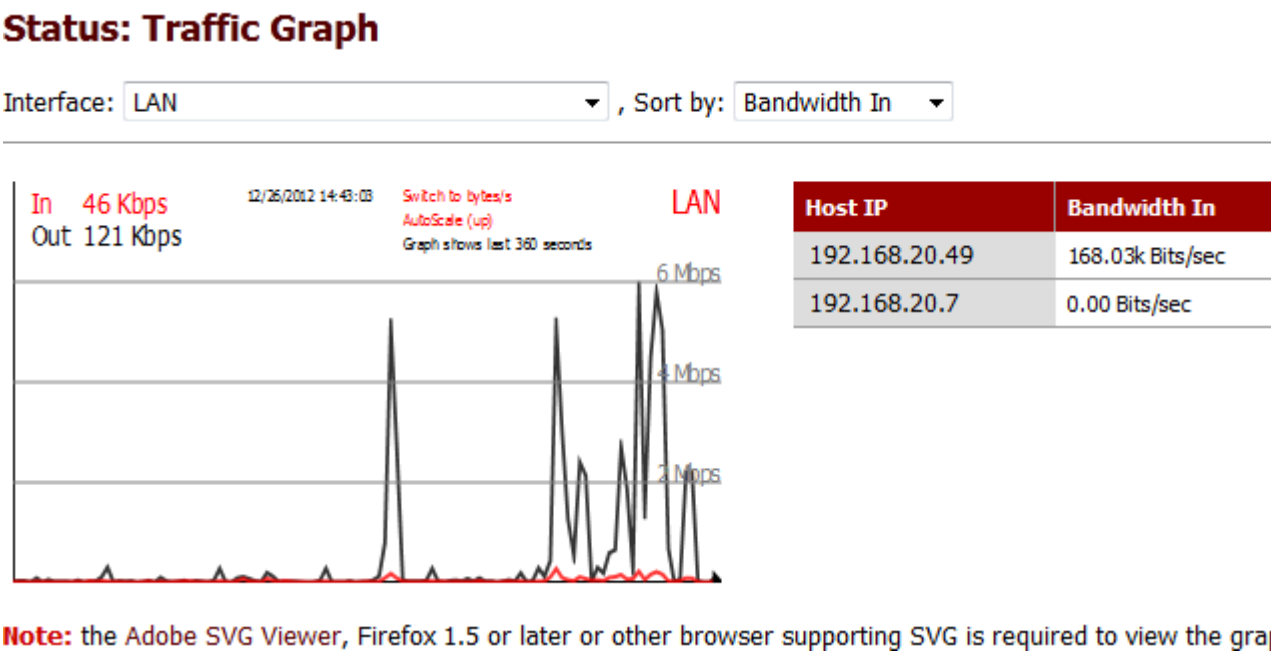
Both the state table and the source tracking table may be reset by going to Diagnostics → States on the Reset States tab. To reset the tables, check either Firewall state table, Firewall Source Tracking, or both, and then click the Reset button.

# Traffic Graphs

Real time traffic graphs drawn with SVG (Scalable Vector Graphics) are available that constantly update. You can find them under Status → Traffic Graphs, and an example of the graph can be found

in Figure 27.7, “Example LAN Graph”. These will allow you to see traffic as it happens, and give a much clearer view of what is happening "now" than relying on averaged data from the RRD graphs.

**Figure 27.7. Example LAN Graph**




Only one interface is visible at a time, and you can choose which one to view from the Interface drop-down list. Once an interface is chosen, the page will automatically refresh and start displaying the new graph. The Dashboard feature in pfSense 2.0 enables the simultaneous display of multiple traffic graphs on a single page.

A table containing momentary glimpses of data being transferring from specific IPs 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.

# System Activity (Top)

From Diagnostics → System Activity, you can view a list of the top active processes in the GUI. This is equivalent to running the command **top -SH** at the shell, except the GUI version does not have the CPU usage summary. Using this view, you can easily see what processes are consuming 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 you're trying to push more than the hardware can handle. If the top process is a PHP process, there could be a process gone astray and not a hardware limitation.

# pfInfo

Diagnostics → pfInfo displays some statistics and counters about how the packet filter is behaving and processing data. The information shown on the page contains items such as:

- Bytes in/out.
- Packets in/out and passed/blocked.
- State table entry count, search rate, insertion rate, removal rate.
- Source tracking entry count, search rate, insertion rate, removal rate.

- Counter statistics for various types of special packets.
- Counters for packets dropped due to exceeding limits such as max states per IP.
- State table max size, source node table size, frag table size, number of allowed tables, and maximum number of table entries.
- State timers for TCP, UDP, and other connections.
- Per-interface packet counters.

# S.M.A.R.T. Hard Disk Status

Starting with pfSense 2.0, the firewall can monitor the health of hard drives that support S.M.A.R.T. monitoring. 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's still useful to identify many kinds of disk failures. Support for S.M.A.R.T. varies by drive and BIOS, but it is fairly well supported in modern ATA drives. S.M.A.R.T. may need to be enabled in the BIOS and on the drive.

Diagnostics → SMART Status allows you to obtain information from the drive, perform or abort drive tests, and view drive logs. In every section of the page, you must first select a Device upon which to act before choosing an option.

## Viewing Drive Information

To view information about a drive, select the Device you want to view, the type of information, and then press View.

### Info

The Info option shows information about the drive itself, including the make, model, serial number, and other technical information about the drive's capabilities, connection, and operation.

```
Model Family:     Hitachi Travelstar 5K500.B
Device Model:     Hitachi HTS545050B9A300
Serial Number:    090630PB4400XXXXXXXX
LU WWN Device Id: 5 000cca 597d1XXXX
Firmware Version: PB4OC64G
User Capacity:    500,107,862,016 bytes [500 GB]
Sector Size:      512 bytes logical/physical
Rotation Rate:    5400 rpm
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:    Thu Dec 27 10:53:54 2012 EST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

### Health

The Health option gives a very brief pass/fail status of the drive, as shown here:

```
SMART overall-health self-assessment test result: PASSED
```

### SMART Capabilities

The SMART Capabilities choice gives a report about what features and tests the drive is capable of, 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 complete
                                        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 sup
                                        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 for most people, 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.

```
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_
  1 Raw_Read_Error_Rate     0x000b   100   100   062    Pre-fail  Always
  2 Throughput_Performance  0x0005   100   100   040    Pre-fail  Offline
  3 Spin_Up_Time            0x0007   138   138   033    Pre-fail  Always
  4 Start_Stop_Count        0x0012   100   100   000    Old_age   Always
  5 Reallocated_Sector_Ct   0x0033   100   100   005    Pre-fail  Always
  7 Seek_Error_Rate         0x000b   100   100   067    Pre-fail  Always
  8 Seek_Time_Performance   0x0005   100   100   040    Pre-fail  Offline
  9 Power_On_Hours          0x0012   099   099   000    Old_age   Always
 10 Spin_Retry_Count        0x0013   100   100   060    Pre-fail  Always
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always
191 G-Sense_Error_Rate      0x000a   100   100   000    Old_age   Always
192 Power-Off_Retract_Count 0x0032   100   100   000    Old_age   Always
193 Load_Cycle_Count        0x0012   100   100   000    Old_age   Always
194 Temperature_Celsius     0x0002   152   152   000    Old_age   Always
196 Reallocated_Event_Count 0x0032   100   100   000    Old_age   Always
197 Current_Pending_Sector  0x0022   100   100   000    Old_age   Always
198 Offline_Uncorrectable   0x0008   100   100   000    Old_age   Offline
```

```
199 UDMA_CRC_Error_Count    0x000a   200   200   000    Old_age   Always
223 Load_Retry_Count        0x000a   100   100   000    Old_age   Always
```

There is a thorough article on Wikipedia for S.M.A.R.T. [http://en.wikipedia.org/wiki/S.M.A.R.T.] that includes a guide for interpreting the values. Some values are more obvious than others, such as 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 drive's overall age and usage. 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 you may want to plan on replacing the disk. 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 bit of a special one, 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. Starting with pfSense 2.0.x, we attempt to disable the hard drive's power management at boot time because otherwise the drive could fail prematurely after running this count up high. You can sometimes hear this cycling happening on drives as a soft clicking noise.

Not shown above, but some drives support other metrics that can be helpful. In particular, some SSDs can give an estimate of their remaining lifetime, in place of the other values that do not apply to an SSD.

### All

Selecting All will, as the name implies, show you all of the information above, and also includes the drive's logs.

## Drive Self-tests

To perform a test on a drive, select the Device you want to test, the type of test, and then press Test.

### 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 SMART Capabilities. An offline test will also cause the drive to update several of the S.M.A.R.T. attributes to indicate the results, so after running a test and checking the results, it is also suggested that you review the S.M.A.R.T. Attributes again as well as the Error log.

### Short

This test takes somewhere in the neighborhood of ten minutes, and checks the drive's mechanics and reading performance. A more accurate estimate of the length the test will take on your drive can be seen in the SMART Capabilities. To see the results of this test, view the Self-test Logs. It can be run at any time and should not impact performance.

### Long

This test is similar to the Short test but it done more thoroughly. The time can depend on the size of the disk, but it would be much, much longer than the short test on its own. A more accurate estimate

of the length the test will take on your drive can be seen in the SMART Capabilities. As with the short test, the results end up in the Self-test Logs.

### Conveyance

This test is not supported by all drives. It's primary purpose is to test the drive after it has been physically relocated to determine if any components have been damaged by the move. It should only take a few minutes to complete. To determine if your drive supports a conveyance test, refer to the SMART Capabilities output.

### Canceling Active Tests

To cancel an active test on a drive, go to the Abort Tests section of the page, select the Device you want to cancel the test on, and then press Abort. Any active tests on the drive will be stopped.

## Drive Logs

To view drive logs, select the Device you want to view, the type of information, and then press 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, etc. 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 the last twenty or so 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 the test, the first logical block address (LBA) is printed to help determine where in the disk the problem lies.

# SMTP and Growl Notifications

You can monitor the status of the system passively using SMTP or Growl notifications to receive alerts about system events. More information about these, including how to configure them, may be found in the section called “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 somewhat static, as with the bogons list or aliases, or dynamic for things like snort or IPs exceeding connection limits. The contents of these tables can be viewed at Diagnostics → Tables. On that page, you select the desired table from the Table drop-down, and the contents are shown. Tables may contain both IPv4 and IPv6 addresses, and the appropriate addresses are used based on the rules in which the tables are referenced.

Individual entries may be removed by clicking the  at the end of their row. Be aware though that tables which are defined manually or by a file will be refreshed when the system performs a filter reload, so it's best to edit an alias and remove an entry rather than removing it from here. Removing entries is best used for dynamic tables like **`snort2c`** (Where blocked offenders from Snort go) or **`virusprot`** (where limit exceeders go) so you can clear an entry out before it automatically expires.

If you have any interface with Block Bogon Networks configured, the **`bogons`** and **`bogonsv6`** tables will show in this list. Along with the contents, there is a Download button that will immediately re-fetch the bogons data rather than waiting for the usual monthly update.

When using automatic outbound NAT, the **tonatsubnets** table shows you the list of networks for which automatic outbound NAT is being performed. It can be handy to look at to confirm tricky NAT issues to confirm that a specific subnet is getting automatic outbound NAT applied.

If you use hostnames in aliases so that they are populated from DNS, viewing the resulting table here also lets you confirm what IPs are actually in the table.

# Testing DNS

Diagnostics → DNS Lookup lets you run simple forward and reverse DNS queries to obtain information about an IP address or hostname, and also to test your DNS servers. To use it, visit the page and enter a Hostname or IP in the box, then press DNS Lookup.

The results of the DNS query are displayed on the page, along with some supporting information and options. For starters, the addresses returned by the DNS query are printed next to the input field. Directly underneath the query results is a link labeled Create alias out of these entries, which does exactly that: It creates an alias under Firewall → Aliases containing the results of the query as the hosts in the alias.

Underneath the results, there is a table showing the Resolution time per server. This lets you see how fast each of your configured DNS servers responded to the specified query, or if they never responded.

In More Information, there are links to pfSense's ping and traceroute functions for this host, and links to external tools for looking up information about who owns the host or IP address.

In other places around the GUI, such as the firewall logs, you may see a icon next to an IP address or a hostname. Those links connect you back to this DNS Lookup page to resolve the address or obtain more information.

# Testing a TCP Port

Diagnostics → Test Port will run a simple TCP port connection test to see if you can communicate from the firewall to another computer or server. This test allows you to determine 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, fill in the fields on the page and then click Test. The Host and Port fields are required, the rest are optional.

| | |
|---|---|
| Host | This is the IP address or hostname of the target system. You must enter a valid value here. |
| Port | This is the TCP port on the target host that you wish to test. This must be a valid port number, meaning an integer between 1 and 65,535. |
| Source Port | If needed, you may specify the source port of the query here. This is not required in most cases, but some rare applications or firewalls do expect a specific source port. |
| Show 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. |
| Interface | If you need to specify a certain source IP address or IP Alias/CARP Virtual IP, it may be chosen here. The service you are testing may require a specific source IP, 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 connecting to a given hostname. If you force IPv4 or IPv6 and use a hostname that does not contain a result using that protocol, it will result in an error. For example if you force IPv4 and use a hostname that only returns an AAAA IPv6 IP address, it will not work. |

# Chapter 28. Packages

The pfSense package system provides the ability to extend pfSense without adding bloat and potential security vulnerabilities to the base distribution. Packages are only supported on full installs and NanoBSD-based embedded installs, not the live CD and older embedded platforms. Embedded versions of pfSense 1.2.3 and later, which are based on NanoBSD, have the capability of running some packages. To see the packages available, 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 allows you to automatically create VPN configuration files.

### Note

These pfSense packages are different than the FreeBSD Ports packages which are covered in the section called “Using Software from FreeBSD's Ports System (Packages)” in the Third Party Software chapter.

By far the most popular package available for pfSense is for 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 lets you view 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 like BandwidthD, NTOP, and Darkstat.
- Extra services like a DNS server, TFTP server, and FreeRADIUS.
- Proxies for other services like SIP and IMSpector, and reverse proxies for HTTP or HTTPS like HAProxy and Varnish.
- System utilities like NUT for monitoring a UPS and LCDProc for using an LCD.
- Popular third-party utilities like nmap, iperf, and arping.
- BGP Routing, OSPF routing, Cron editing, Nagios and Zabbix agents, and many, many others.
- Some items that were formerly in the base system but were moved to packages, such as RIP (routed) and OLSRD

As of this writing there are more than 50 different packages available; too many to cover them all in this book! If you would like to see the full list, it will be available from within your pfSense system by browsing to System → Packages.

You may notice that the packages screen may take a little longer to load than other pages in the web interface. This is because it fetches the XML package information from our servers before the page is rendered to provide the most up to date package information. If your firewall does not have a functional Internet connection including DNS resolution, this will fail and notify you, as in Figure 28.1, “Package information retrieval failed”. If you have previously successfully retrieved the package information, it will be displayed from cache, but you may not have the most recent information. 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 servers assigned by your ISP are functioning. You may wish to override these dynamically assigned servers with OpenDNS [http://www.opendns.com] or another DNS server.

**Figure 28.1. Package information retrieval failed**


A growing number of packages have a Package Info link in the package list, pointing to a site with more information on that specific package. You should read the information in the Package Info link before installing a package. After installation, you can find the most recent Package Info link for each installed package on the Installed Packages tab.

# pfSense Package Format

In pfSense 2.0.x and before, we used the same package format as FreeBSD for binaries. This was convenient, but it also led to situations where packages could install on top of items in the base system and then when they were removed, it could break the pfSense installation in various ways. In some rare cases it could render the GUI inoperable.

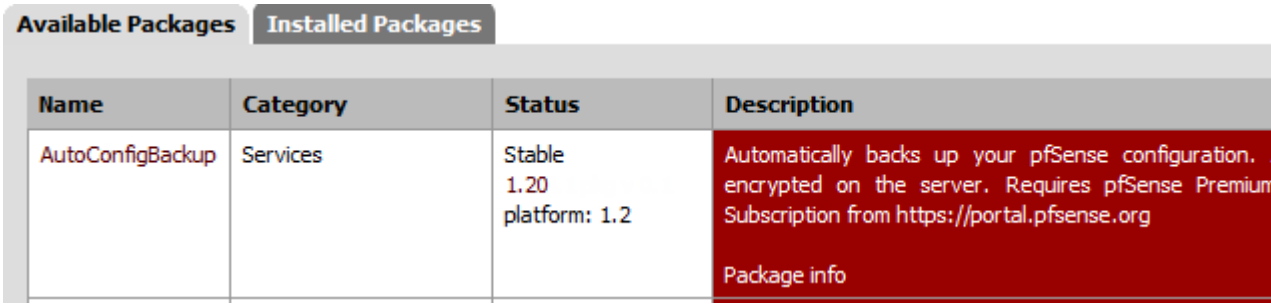
Starting with pfSense 2.1, we switched to using the package format employed by PC-BSD, called Push Button Installer or PBI for short. Each PBI is a self-contained file that installs all of the needed dependencies for a program inside an isolated directory. This solves the problem of packages needing to conflict with the base system since they don't touch the base system in that way. The downside is that they do take up more space, and the package code is a little more complicated.

To the end user, there isn't much visible about this change. The package management in the pfSense GUI works the same as it always has.

# Installing Packages

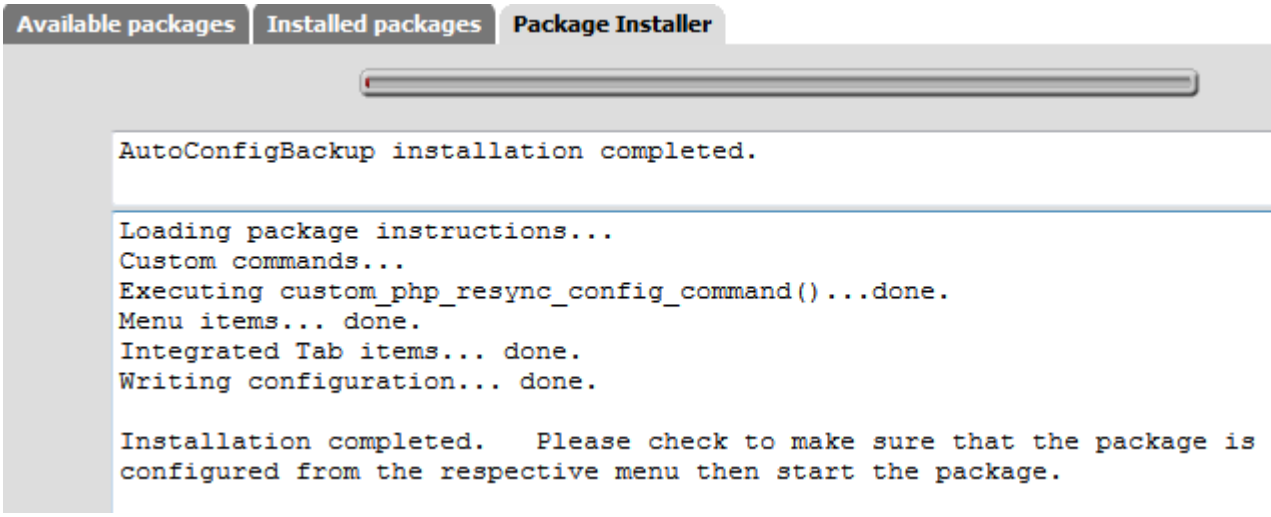
Packages are installed from System → Packages. The listings there, exemplified by Figure 28.2, “Package Listing”, will show a package's name, category, version and status, a package information link, and a short description. Pay very close attention to Status before installing packages, some packages are experimental and should never be installed on critical production systems. You should also keep the installed packages to the bare minimum required for your deployment. The packages are presented in alphabetical order.

**Figure 28.2. Package Listing**




Packages are installed by clicking the button to the right of their entry. Upon clicking , you will be taken to the package installation screen where the install progress will be displayed (Figure 28.3, “Post-Install Package Screen”).

**Figure 28.3. Post-Install Package Screen**

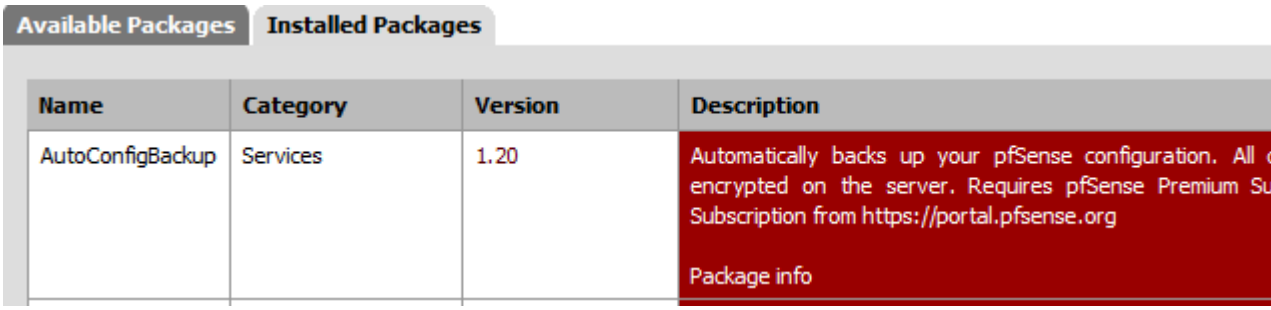



# Reinstalling and Updating Packages

Packages are reinstalled and updated the same way. Start by going to System → Packages, and clicking the Installed packages tab. The listings there should look like Figure 28.4, “Installed Package List”. Find the package you want to reinstall or update in the list. If there is a newer version available than you have installed, the Package Version column will be highlighted in red stating the old and new versions. Click to update or reinstall the package.

Another reinstallation choice would be to reinstall just the XML GUI components of a package, which can be done by clicking next to the package entry. Unless instructed to do so by a developer, you shouldn't use this option as it can miss updates to the binaries that the latest GUI components may require.

**Figure 28.4. Installed Package List**




# Uninstalling Packages

To uninstall a package, browse to System → Packages, and click the Installed packages tab. Find the package in the list, and click the button. The package will then be removed from the system.

Some experimental packages overwrite files distributed with the base system. These packages cannot be uninstalled, as doing so would break the remaining base system. The package entry may still show the uninstall icon, but they will still be present after their attempted removal. Packages with this quirk will be labeled as such in their description field. If you upgrade the system, it will overwrite the changes made by these packages, so this is a possible means of removing them. Be very careful with any packages that cannot be uninstalled, they are generally meant for experimentation on non-critical systems.

# Developing Packages

Packages are relatively simple to develop, and you may find that either you or your organization may benefit from developing a package that does not exist. For those interesting in creating their own packages, resources are available on the pfSense Documentation Wiki [http://doc.pfsense.org/index.php/Developing_Packages]. If you create a package and think it may be of use to others, contact us and your work can be evaluated for inclusion into the package system for everyone to see.

# A Brief Introduction to Web Proxying 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 proxying software, but a quick run-through to get it up and running and to cover the most commonly asked questions about their capabilities. In this section we will cover Squid 2.x 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.

There are other, similar topics using additional packages that are available such as Dansguardian, Squid 3.x, HAVP, and Sarg. Though we will not cover them in this book, you can find more information on our documentation wiki and on the forum.

This discussion assumes you have a simple single LAN and single WAN configuration. If you encounter problems, visit the documentation wiki and forum for guidance and assistance.

## Squid — Caching Web Proxy

Squid is the foundation of many other tasks that start with proxying: It can act as a cache for improving web performance, it can hook into SquidGuard, Dansguardian, or HAVP 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 the package is installed, visit Services → Proxy Server to configure Squid. The Squid configuration is broken up into several tabs. Before leaving a tab, click Save if you have made any changes.

On the General tab, select **LAN** for the Proxy Interface. Check Allow users on interface, Transparent Proxy, and Bypass proxy for Private Address Space.

If you need certain local client IPs to bypass the proxy, put them in Bypass proxy for these source IPs separated by a semicolon. If you need certain remote servers to bypass the proxy as users attempt to access them, put them in Bypass proxy for these destination IPs separated by a semicolon.

If pfSense is running on a full install (NOT embedded/NanoBSD) and web access reporting is desired, check Enable Logging.

Change the Visible Hostname and Administrator E-mail to reflect the proper values for your firewall and a usable contact address. If a user encounters a proxy error, these will be shown to the user so they may contact you for support.

On the Cache Mgmt Tab, change the Cache Size to a value that is reasonable for your available drive space and RAM. If running NanoBSD, enter *0* here and set the Hard Disk Cache System to **null**. Other parameters on the page 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.

If you have more subnets behind a static route on the LAN, be sure to visit the Access Control tab and add them into the Allowed Subnets list.

The proxy should be up and running now. If using transparent mode, loading a proxy test site such as http://www.lagado.com/proxy-test should now reveal that you the request was routed through a proxy.

# SquidGuard — Web Access Control and Filtering

The SquidGuard package enabled 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 documentation wiki and forum for more information and related tutorials.

If you would like to use this, first install and configure Squid as described in the previous section, then install the SquidGuard package, and finally visit Services → Proxy Filter to configure SquidGuard.

## General Settings

First, check Enable on the General Settings tab to enable SquidGuard.

If SquidGuard block event logging and GUI event logging are desired, check the relevant boxes in the Logging Options section.

### Note

After saving the settings on any tab in SquidGuard, one must always return to the General Settings tab and press 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. Blacklists do not work properly on NanoBSD. If you want to use blacklists, check Blacklist and fill in a Blacklist URL. The two most common lists are the MESD list[1] and the Shalla list[2].

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.

## 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, you must fill in a few options:

| | |
|---|---|
| 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 that you would like to block, such as *`www.facebook.com`*, *`google.com`*, *`microsoft.com`*, etc. Multiple domains may be entered, separated by either a space or a newline. |
| Redirect mode | The Redirect mode 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 here is **`int error page`**. |

[1] http://squidguard.mesd.k12.or.us/blacklists.tgz
[2] http://www.shallalist.de/Downloads/shallalist.tar.gz

| | |
|---|---|
| Redirect | If the user is redirected using **`int error page`**, you can enter the error message for the user to see in this box. If you use an external redirect type, enter a full URL (including `http://` or `https://`) to the desired target site. |

## Access Lists (ACLs)

There are two types of ACL entries in SquidGuard: The Common ACL, which is the default ACL applied to all users, and Group ACL entries which are applied to specific IPs, groups of IPs, or Networks.

First, visit the Common ACL tab. Here you can choose the default actions of all available categories from blacklists or those defined locally. To do this, click Target Rules List (click here), 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. Create a new entry, give it a Name, and in the Client (source) box, enter the user's IP address, subnet, etc. Multiple values can be entered, separated by spaces. Now click Target Rules List (click here) and define the list of actions for this specific set 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.

After saving the settings, return to the General Settings tab and press 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. This feature is not compatible with embedded/NanoBSD installs without manual modifications that are beyond what most users are capable of performing. After the Lightsquid package has been installed, the report settings may be found under Status → Proxy Report.

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 press Refresh Full to build the initial report. Wait a few minutes, then click the Lightsquid Report tab to view the report.

If there is no data in the report, check to make sure that you have set Enable Logging in Squid, and that the user traffic is actually going through the proxy as you expect.

# Transparent Proxying and HTTP/HTTPS

When using a proxy, it is possible to intercept HTTP traffic transparently. That is, you can grab it automatically and force it trough 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 made 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.

There are proxy packages that are 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 either 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 our documentation wiki, our forum, and elsewhere around the web.

# Chapter 29. Third Party Software and 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.

# RADIUS Authentication with Windows Server

Windows 2000 Server and Windows Server 2003 can be configured as a RADIUS server using Microsoft's Internet Authentication Service (IAS). This allows you to authenticate the OpenVPN, pfSense PPTP server, Captive Portal, or PPPoE server from your Windows Server local user accounts or Active Directory. In Windows 2008, this was changed to use the Network Policy Server (NPS), and functions quite a bit differently. The previous copy of this book covered IAS for Windows 2000 and Server 2003, now we'll cover NPS for the more recent Microsoft Server operating systems.

## 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 is also possible to install it on a member server, which may be desirable in some environments to reduce the attack footprint of your domain controllers — each network-accessible service provides another potential avenue for compromising your server. NPS does have a solid security record, especially compared to other things that must be running on your 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 the Windows Server, go to Server Manager, then click on Roles on the left and expand it, then click Add Roles on the right. On Server 2012, from the System Manager Dashboard, click Add Roles and Features. The Add Roles wizard will appear, click Next to skip the intro screen. On Server 2012, you'll need to click past Role-based or feature-based installation and Next once more, then select your server from the list, and click Next again.

Now you will be presented with a list of roles that can be added to your server. From the list, check Network Policy and Access Services. If presented with a screen asking for additional features, click Add Features. Now click Next on each screen until you reach the end of the wizard, then click Finish or Install, depending on your version.

## Configuring NPS

To configure NPS, bring up the Server Manager and you should now see Network Policy and Access Services (2008) or NAP (2012).

First a RADIUS client will be added for pfSense, then remote access policies will be configured.

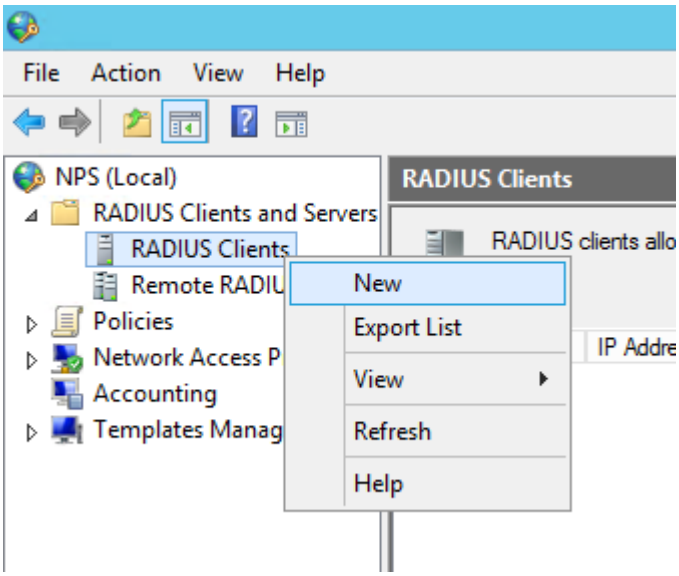
### Adding a RADIUS Client

You'll need to open up the NPS configuration first. On 2008, it appears in the server manager tree and you keep expanding the view under it until you see RADIUS Clients and Server, then RADIUS

Clients. On 2012, from the Server Manager dashboard, click NAP, then from the server list, right click on your server and click Network Policy Server, then expand RADIUS Clients and Server, then RADIUS Clients.

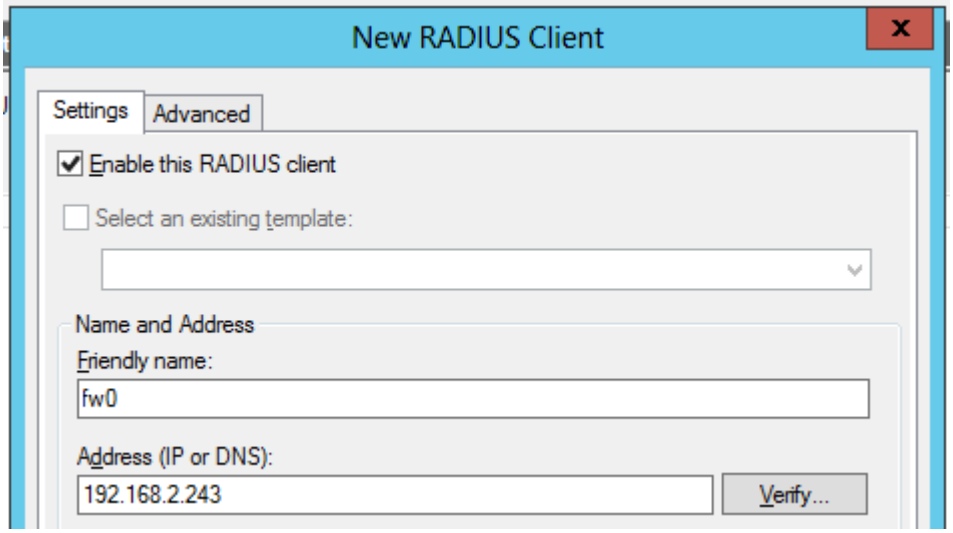
Right click on RADIUS Clients and click New, as shown in Figure 29.1, “Add new RADIUS client”.

### Figure 29.1. Add new RADIUS client



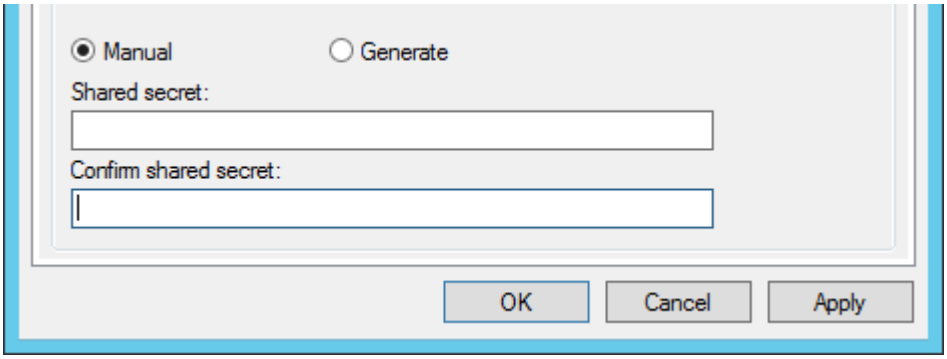

Enter a Friendly name for your firewall, like shown in Figure 29.2, “Add new RADIUS client — Address”, which can be your hostname or FQDN. The Address (IP or DNS) field must be the IP address that pfSense will initiate its RADIUS requests from, or a FQDN that will resolve to that IP address. This will be the IP address of the interface closest to the RADIUS server. If the RADIUS server is reachable via your LAN interface, this will be the LAN IP. In deployments where pfSense is not your perimeter firewall, and your WAN interface resides on the internal network where your RADIUS server resides, the WAN IP address is what you must enter here.

### Figure 29.2. Add new RADIUS client — Address



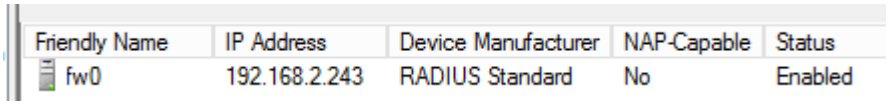

Next you'll need to fill in a shared secret, as shown in Figure 29.3, “Add new RADIUS client — Shared secret”. This shared secret is what you will enter on pfSense later. You can have Windows create one automatically for you by clicking Generate. Once you have entered a Shared secret that you are happy with, Click OK.

**Figure 29.3. Add new RADIUS client — Shared secret**




Now you have completed your NPS configuration. You can see the RADIUS Client you just added as in Figure 29.4, “Listing of the RADIUS Client”.

**Figure 29.4. Listing of the RADIUS Client**




Now you are ready to configure pfSense with the RADIUS information configured here, using the IP address of the NPS server, and the shared secret configured previously. Refer to the portion of this book describing the service you wish to use with RADIUS for more guidance. RADIUS can be used in the User Manager (Chapter 7, *User Management and Authentication*), for Captive Portal (the section called “Portal Configuration Using RADIUS Authentication”), the PPTP server (the section called “Authentication”), and the PPPoE server (the section called “PPPoE Server”), and also in some packages.

# Configuring Users and Network Policies

Whether a user can authenticate via RADIUS is controlled through Network Policies. Using Network Policies you can simply 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.

After configuring users and remote access policies as desired, you are ready to test the service you are using with RADIUS on pfSense.

## Adding a Network Policy

In the NPS configuration window, expand NPS (Local), Policies, then Network Policies. Right click on Network Policies and click New.

In the policy name, enter *`Allow from pfSense`*, and leave the Type of network access server set to **`Unspecified`**, then click Next.

The next step is the Specify Conditions window. In there, click Add. Select Windows Groups, then click Add again. Now, enter or select the name of your user group which contains your VPN users, e.g. *`VPNUsers`*, then click OK, then Next.

On the next screen of the wizard, choose Access granted, then Next once more.

For Authentication Methods, in addition to the modes that are pre-selected, select both Encrypted Authentication (CHAP) and Unencrypted Authentication (PAP, SPAP), then Next. If you are prompted to view a help topic, decline.

If there are any additional access restraints you would like to configure, add them on this screen, otherwise press Next on the remaining screens until you can click Finish.

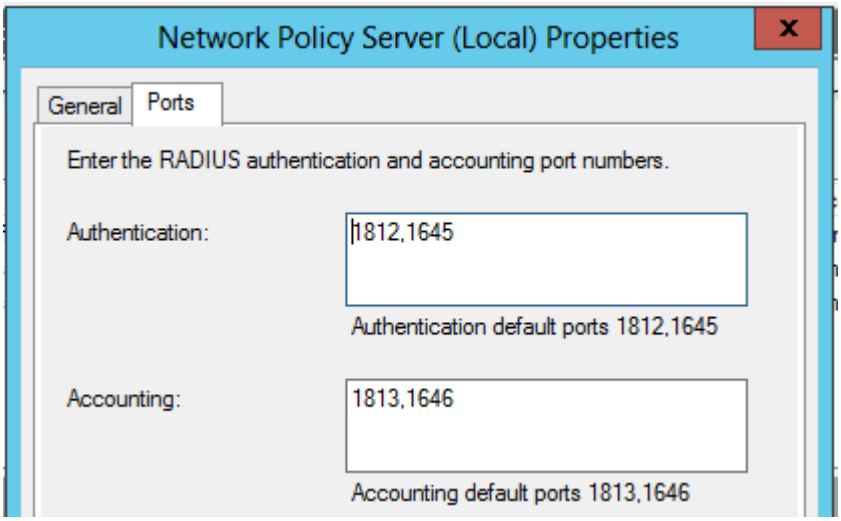
## Troubleshooting NPS

Should authentication fail, this section describes the most common problems users encounter with NPS.

### Verify port

First ensure the default port 1812 is being used. If your NPS server was previously installed, it may have been configured with non-standard ports. In the NPS console, right click on NPS (Local) at the top left of the console and click Properties. Then click the Ports tab. You can specify multiple ports by separating them with a comma (as shown in Figure 29.5, “NPS Ports”). Port *1812* must be one of the ports configured for Authentication. If you are using RADIUS accounting functionality as well, port *1813* must be one of the ports specified in Accounting.

**Figure 29.5. 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 and, if access is denied, the reason it was denied. 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.

# Free Content Filtering with OpenDNS

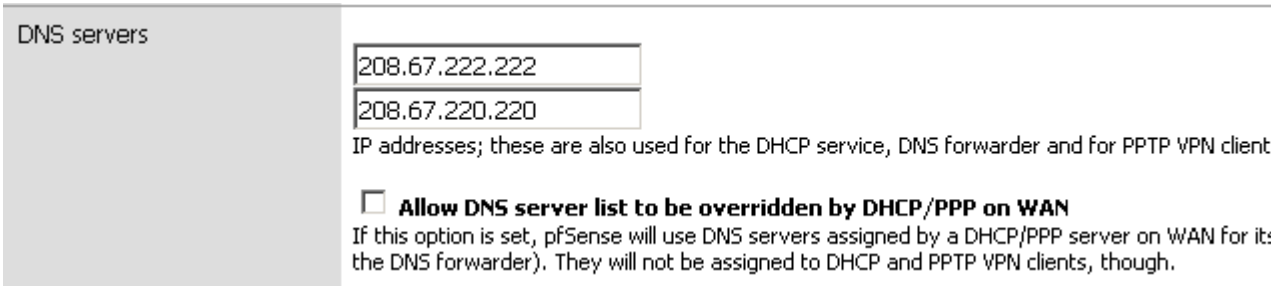
pfSense doesn't include any content filtering software at the time of this writing, but there is a great free option in integrating OpenDNS [http://www.opendns.com]. First you need to configure your network to use OpenDNS's DNS servers for all recursive queries.[1]

[1]Note: I am in no way affiliated with OpenDNS, just a very satisfied user of their services in multiple locations, and I have had numerous people thank me for referring me to them. They truly have an impressive offering.

# Configuring pfSense to use OpenDNS

Visit the System → General Setup page, enter OpenDNS's two DNS servers there, and uncheck the "Allow DNS server list to be overridden by DHCP/PPP on WAN" box (Figure 29.6, “Configuring OpenDNS on pfSense”).

**Figure 29.6. Configuring OpenDNS on pfSense**



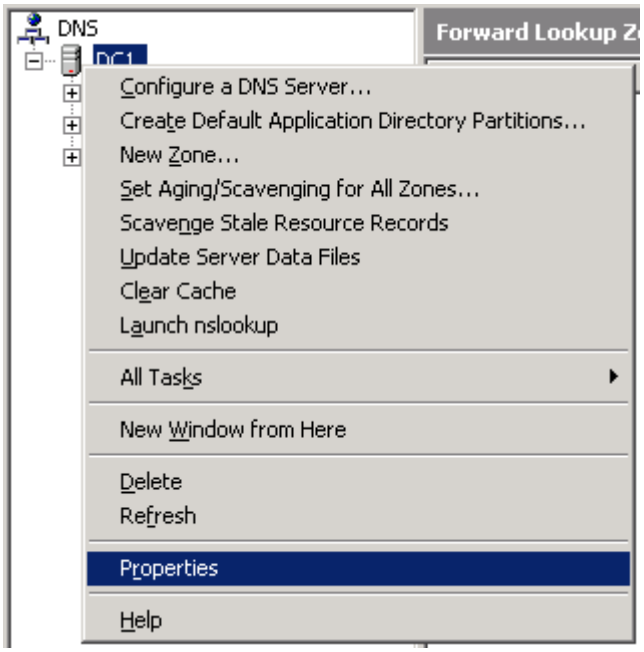

If your internal machines use pfSense's DNS forwarder as their only DNS server, this is all you need to change to use OpenDNS for your name resolution.

# Configure internal DNS servers to use OpenDNS

If your internal machines use an internal DNS server, it needs to be configured to send its recursive queries to OpenDNS's servers. I will explain how to accomplish this with Windows Server's DNS server.

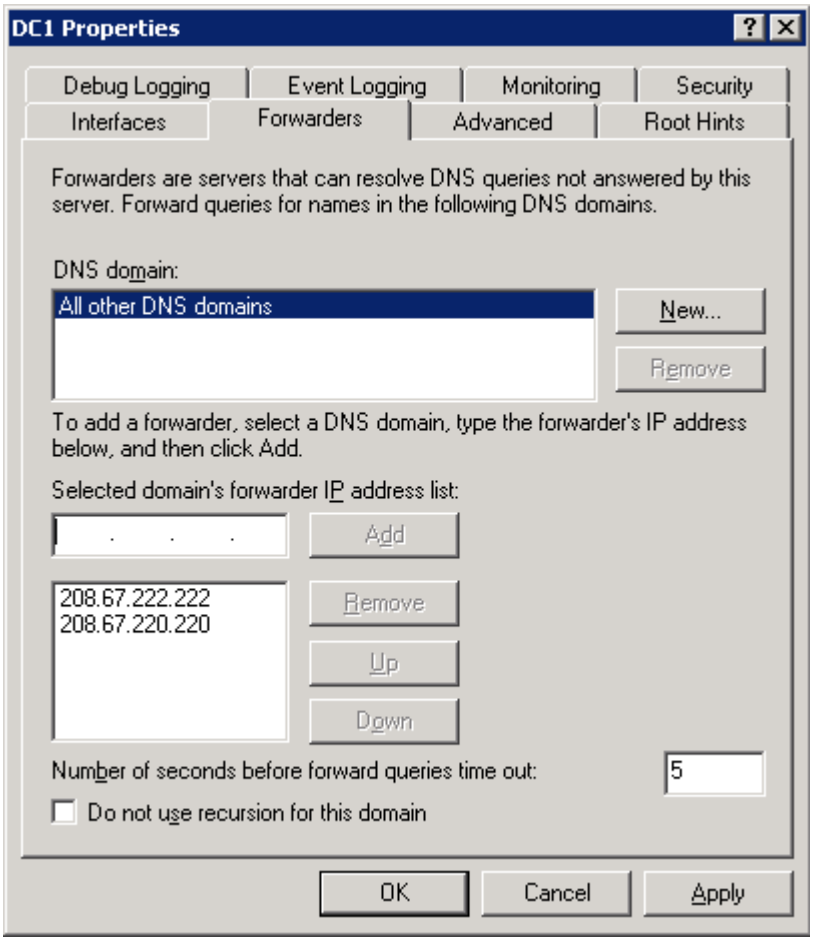
## Configuring Forwarders in Windows Server DNS

**Figure 29.7. Windows Server DNS Properties**




Open the DNS MMC snap-in under Administrative Tools, DNS. Right click on the server's name and click Properties, as shown in Figure 29.7, “Windows Server DNS Properties”.

**Figure 29.8. Windows Server DNS Forwarders**




Select the Forwarders tab, and add OpenDNS's two DNS servers in the forwarder list for "All other DNS domains" as in Figure 29.8, “Windows Server DNS Forwarders”, then click OK.

Then repeat this for each of your internal DNS servers.

# Configuring OpenDNS Content Filtering

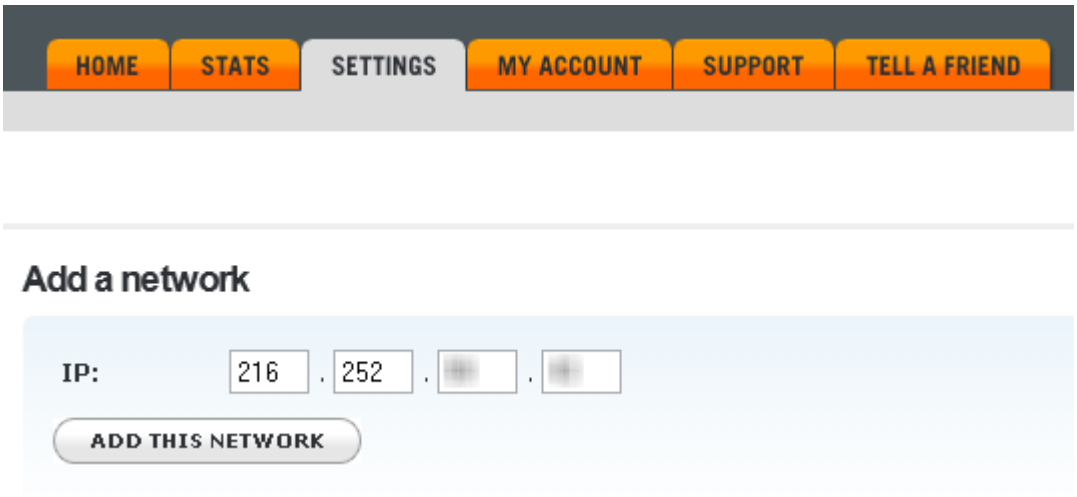
Now you need to configure your content filtering as desired on the OpenDNS site.

## Sign up for an OpenDNS account

Browse to http://www.opendns.com and click the Sign In link. Then click the "Get Started" link and go through the account creation process. They do have several for-pay options currently, but also maintain free accounts for many uses.

## Define your network(s) in OpenDNS

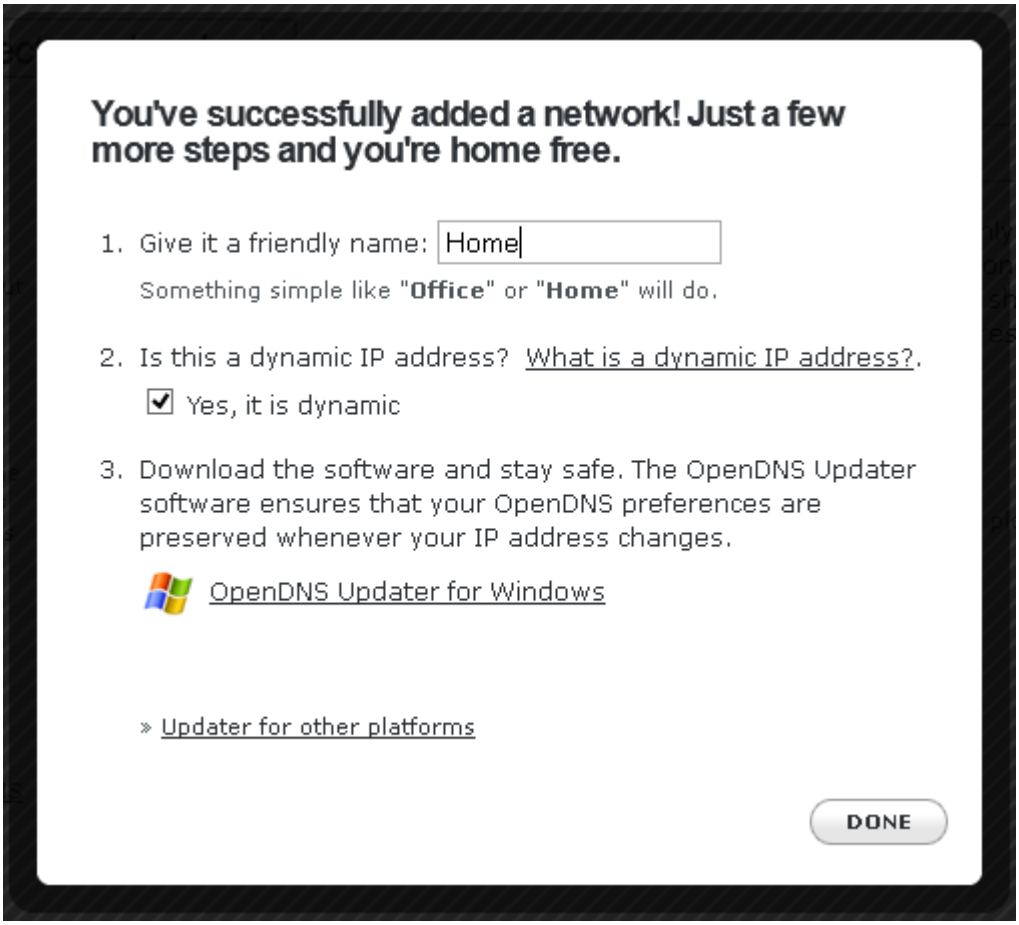
**Figure 29.9. Add a network**




OpenDNS first needs to be able to determine which DNS queries are from your network to be able to filter according to the policies defined in your account. After logging into your OpenDNS account, click the Networks tab (Figure 29.9, “Add a network”). It will automatically show the public IP your HTTPS session is coming from, with a button to add this network to your account. Click the Add this network button.

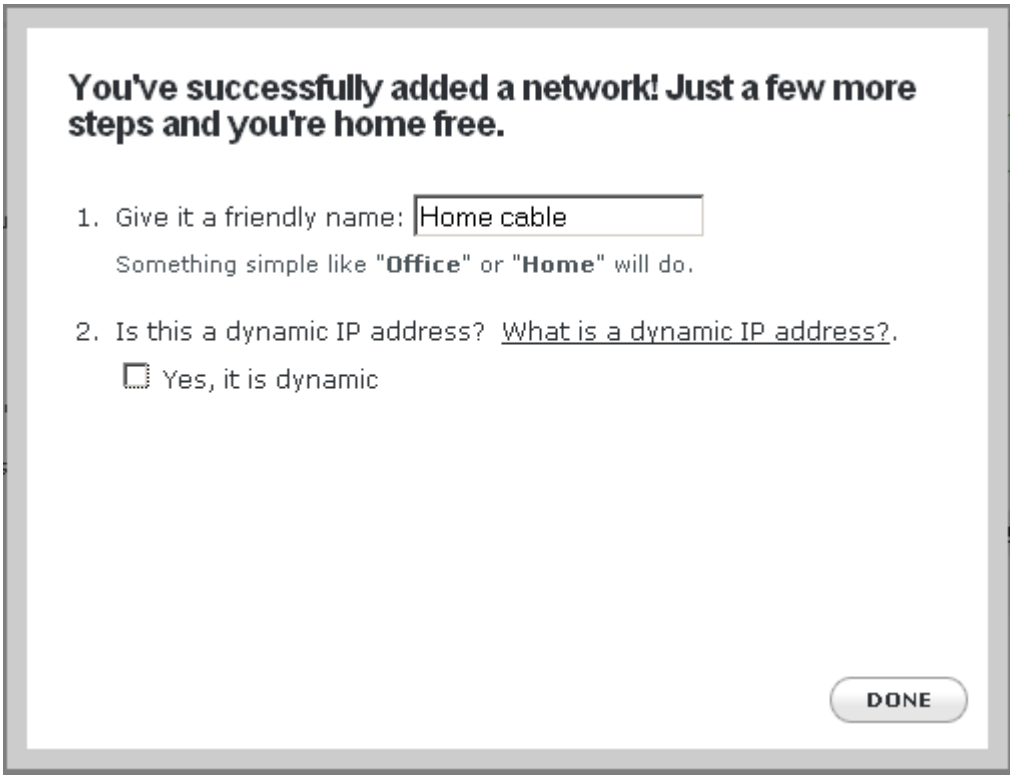
This will bring up a window prompting whether your IP is static or dynamic (Figure 29.10, “Adding a dynamic IP connection”). If you have a dynamic IP connection, you will have to run the OpenDNS Updater for Windows on a machine inside your network to ensure your address is kept up to date with OpenDNS. Your IP address is the only means of identification OpenDNS has of your network. If your IP is not correct in your OpenDNS settings, your content filtering will not function as configured in your account.

**Figure 29.10. Adding a dynamic IP connection**



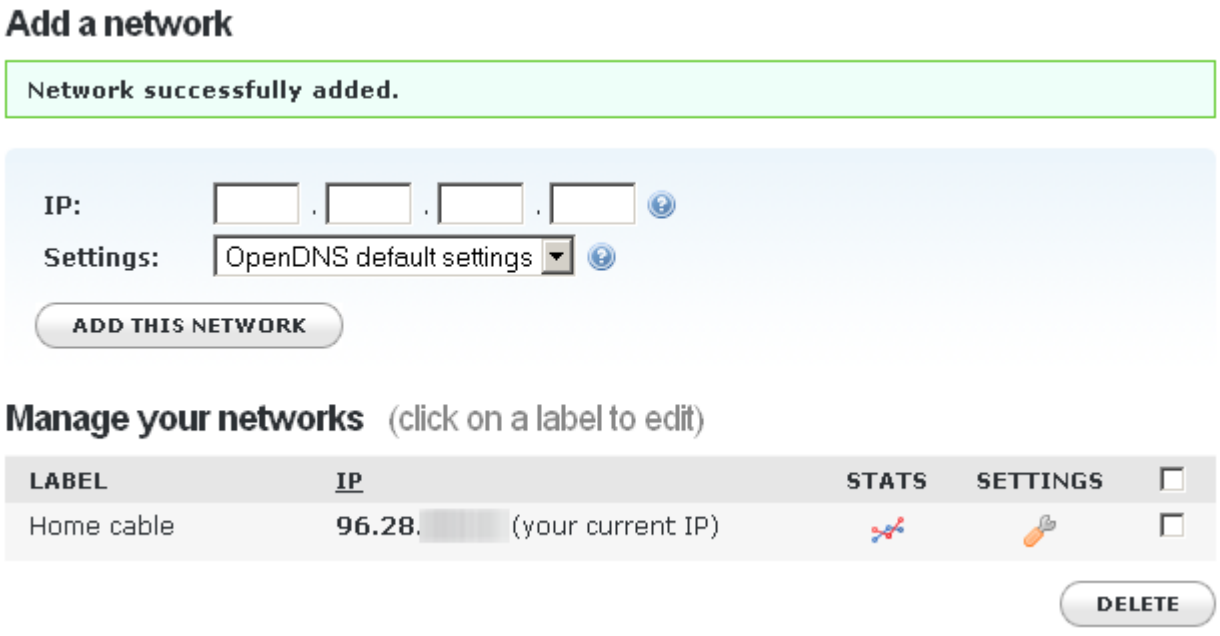

For static IP connections, uncheck the "Yes, it is dynamic" box and give the connection a name (Figure 29.11, “Adding a static IP connection”). For static IP connections, you don't need to run the updater client.

**Figure 29.11. Adding a static IP connection**




After adding your network to your account, you will see it in your network list like that in Figure 29.12, “Network successfully added”.

**Figure 29.12. Network successfully added**

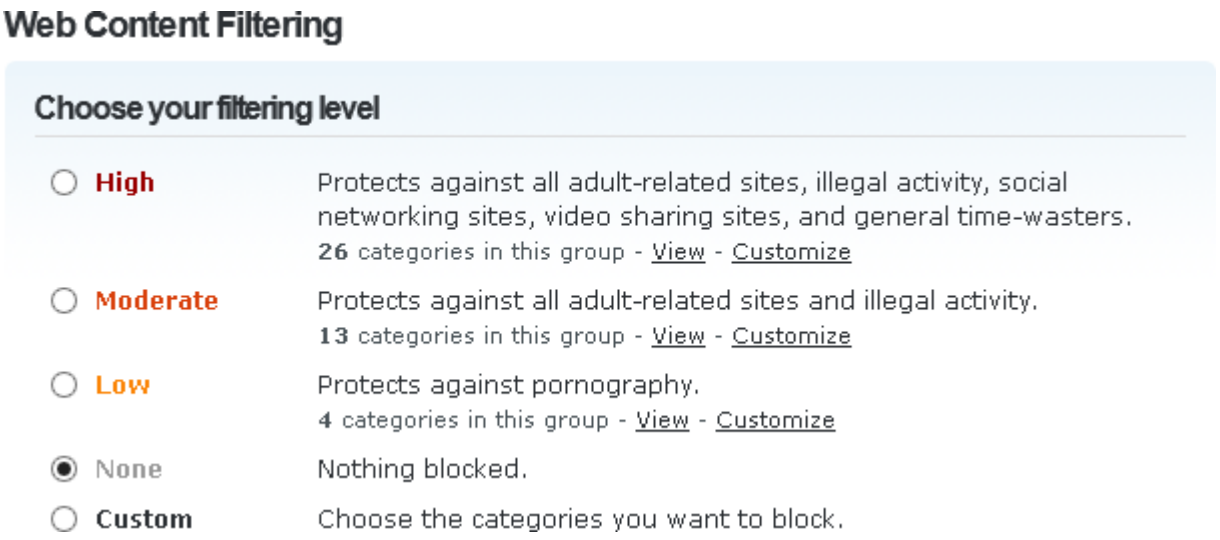



Your network is now ready to use OpenDNS, though you still need to configure your desired content filtering settings. OpenDNS will send you a confirmation e-mail to ensure you used the correct IP address, be sure to follow the instructions in the e-mail.

## Configuring content filtering settings for your account

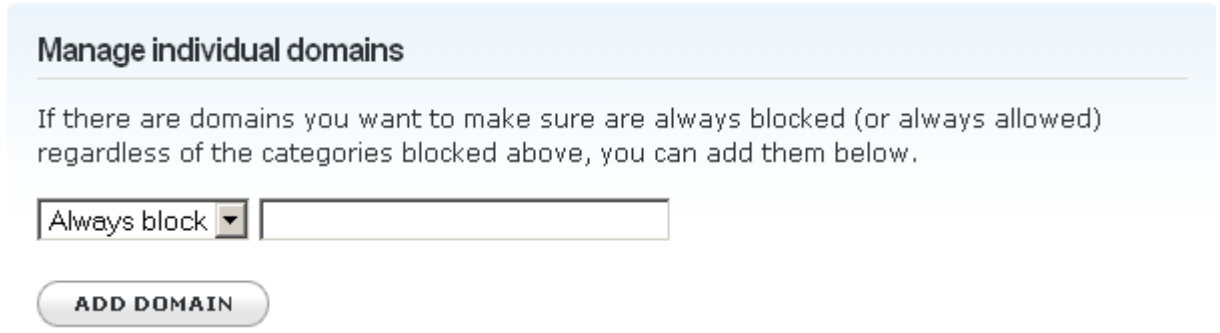
To configure your content filtering settings, click the Settings tab at the top of the OpenDNS website. A list of levels like that in Figure 29.13, “Content filtering level” should appear. You will see your current filtering level is None, which does not block anything. You can select from four different pre-defined filtering levels, or choose Custom and select which categories you wish to block.

**Figure 29.13. Content filtering level**




You can also block or allow specific domains, overriding your overall content filtering configuration, at the bottom of this screen (Figure 29.14, “Manage individual domains”).

**Figure 29.14. Manage individual domains**




OpenDNS provides a number of other configuration settings allowing you great control over DNS for your network. Their site contains a number of knowledge base and support articles detailing some of the possibilities, and all of the functionality is well described throughout the management interface. You don't have to stop at just content filtering — review everything else OpenDNS has to offer, as you may be able to put it to good use. OpenDNS has moved some formerly free options to paid services, but still has quite a few useful free options available.

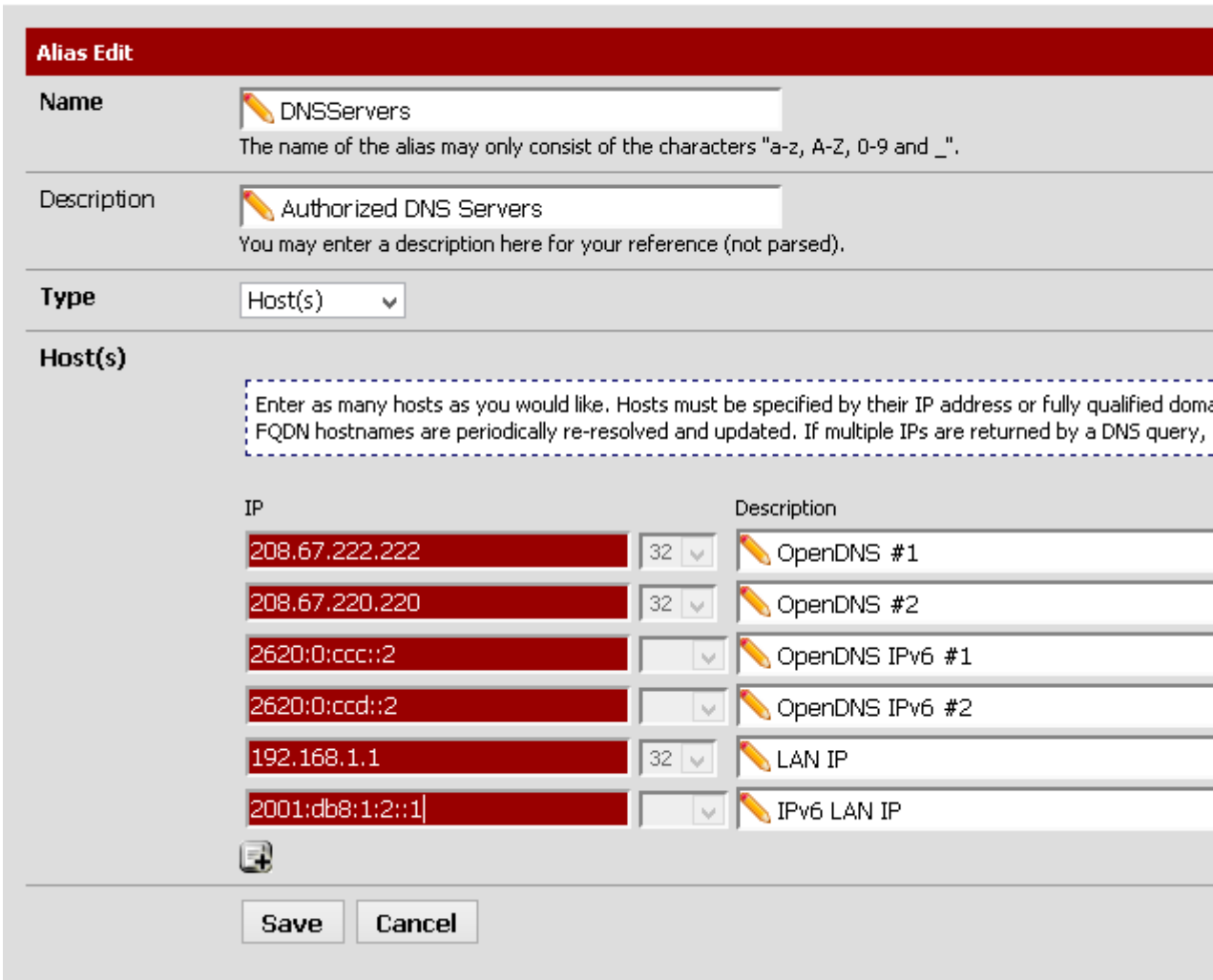
# Configuring your firewall rules to prohibit other DNS servers

Now that your internal systems are all using OpenDNS as their DNS service, you will want to configure your firewall rules so no other DNS servers can be accessed. Otherwise internal users could simply change their machines (if they have the user rights to do so) to use a different DNS server that does not enforce your content filtering and other restrictions.

## Create a DNS Servers alias

First you will want to create an alias containing the DNS servers that internal machines are allowed to query, like the one in Figure 29.15, “DNS servers alias”. The LAN IP is listed because this example network uses the DNS forwarder as its internal DNS server, and this allows DNS queries from the LAN to the LAN IP. It also allows recursive queries from internal DNS servers, and the direct assignment of OpenDNS's DNS servers on internal machines. Note that unless you disable the anti-lockout rule, it isn't necessary to add the LAN IP here, but I recommend adding it regardless for clarity. Refer to the section called “Anti-lockout Rule” for more information.

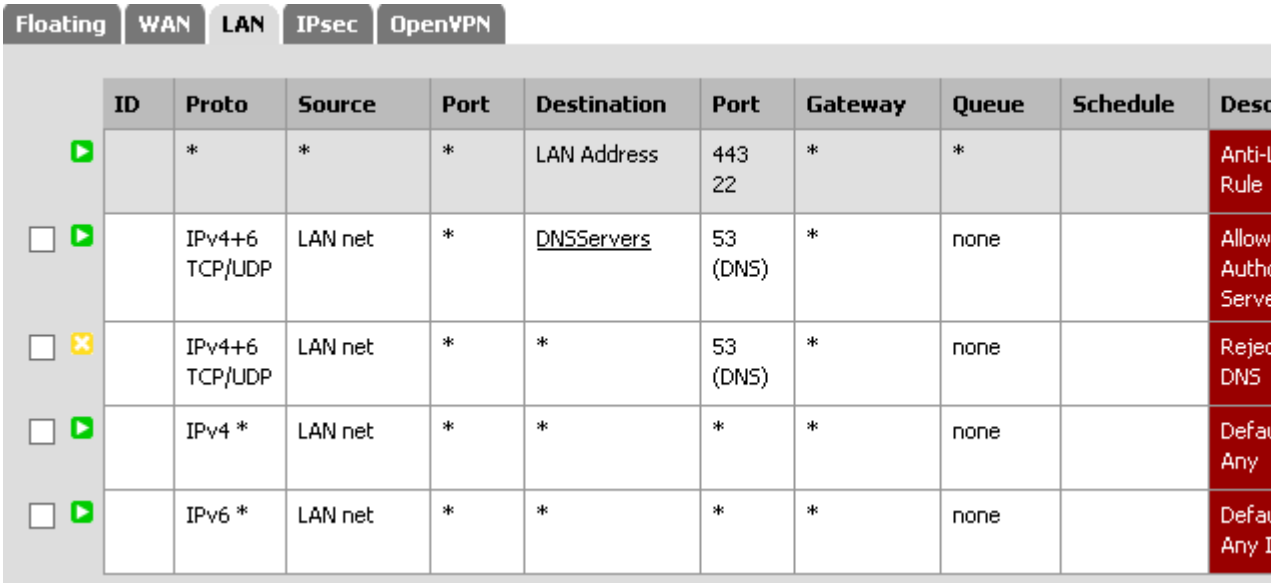
**Figure 29.15. DNS servers alias**




## Configure firewall rules

Now you need to configure your LAN rules to allow DNS destined for the previously created alias, and block DNS to other destinations if any of your other rules would permit DNS, such as the default LAN rule. As discussed in the firewall chapter, I prefer using reject rules for traffic blocked on internal interfaces. The ruleset in Figure 29.16, “LAN rules to restrict DNS” is kept short and simple for the sake of illustration — I recommend significantly stronger egress filtering than this shows, as described in the firewall chapter.

**Figure 29.16. LAN rules to restrict DNS**




## Finishing Up and Other Concerns

And that's it. You now have a free content filtering solution integrated with pfSense in a means that makes it very difficult for the average user to evade. Note it isn't impossible to get around, especially with as permissive of a ruleset as the example above shows. There are several possibilities for tunneling DNS through that ruleset, with VPN connections, SSH port forwarding, and more. But if you allow any traffic through your firewall, that's always going to be a possibility. Properly locked down end user machines in combination with the above provides a strong content filtering solution that's difficult for users to bypass.

# Syslog Server on Windows with Kiwi Syslog

pfSense can send logs to an external server via the syslog protocol (the section called “Remote Logging with Syslog”). For Windows users, Kiwi Syslog Server[2] is a nice free option for collecting logs from your pfSense installs. 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, for a veteran FreeBSD system administrator many familiar FreeBSD packages can also be used. 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 be encountered. pfSense does not include a compiler in the base system for many reasons, and as such software cannot be built locally. However, you can install packages from FreeBSD's pre-built package repository.

## Concerns/Warnings

Before you decide to install additional software to pfSense that is not a sanctioned package, there are some topics that need to be taken into account.

[2] http://www.kiwisyslog.com/

## Security Concerns

Any extra software added to a firewall is a security problem, and should 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 pfSense systems are run on hardware that can handle the traffic load with which they are tasked. If you find that you have horsepower to spare, it may not hurt the system to add additional software. That said, be mindful of the resources that will be consumed by the added software.

## Conflicting Software

If you install a package which 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 you are after does not already exist in the pfSense system 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 are altered to accommodate the methods used by pfSense.

There have also been cases where software has installed additional web pages that are not protected by pfSense's authentication process. Test any installed software to ensure that access is protected or filtered in some manner.

## Lack of Backups

When installing packages in this manner, you must ensure that you backup any configuration or other needed files for this software. These files will not be backed up during a normal pfSense backup and could be lost or changed during a firmware update. You can use the add-on package described in the section called “Backup Files and Directories with the Backup Package” to backup arbitrary files such as these.

# Installing Packages

To install a package, you must first make sure that the proper package site will be used. pfSense is compiled against a specific FreeBSD-RELEASE branch, and the packages there can become stale within a short amount of time. To work around this, specify the path to the set of packages for FreeBSD-STABLE before attempting to install a package:

```
# setenv PACKAGESITE=ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8-sta
# pkg_add -r tcpflow
```

Or you can supply a full URL to a package:

```
# pkg_add -r ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8-stable/Lat
```

The package should download and install, along with any needed dependencies.

It is also possible to build a custom package on another computer running FreeBSD and then copy/install the generated package file onto a pfSense system. Due to the complexity of this topic, it won't be covered here.

# Maintaining Packages

You can view a list of all installed packages like so:

```
# pkg_info
```

To delete an installed package, you must specify its name fully or use a wildcard:

```
# pkg_delete lsof-4.82,4
# pkg_delete tcpflow-\*
```

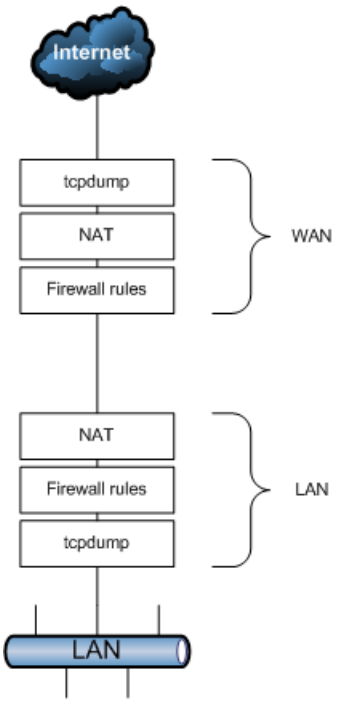
# Chapter 30. Packet Capturing

Capturing packets is the most effective means of troubleshooting problems with network connectivity. Packet capturing (or "sniffing") tools like **tcpdump** show what is "on the wire" — coming in and going out of an interface. Seeing how traffic is received by the firewall and how it leaves the firewall is a great help in narrowing down problems with firewall rules, NAT entries, and other networking issues. In this chapter, we cover 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 what is on the wire. It is the first 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 makes it to that interface on your firewall regardless of whether that traffic will be blocked by your firewall configuration. Figure 30.1, “Capture reference” illustrates where **tcpdump** and also the WebGUI packet capture interface ties into the processing order.

**Figure 30.1. Capture reference**




# Selecting the Proper Interface

Before you can start any packet capture, you need to know from where the capture should be taken. A packet capture will look different depending upon the interface chosen, and in certain scenarios it's better to capture on one specific interface, and in others, running multiple simultaneous captures on different interfaces is preferable. In using **tcpdump** at the command line, you will need to know the "real" interface names that go with the friendly names shown in the WebGUI. You may recall these from when the interfaces were originally assigned, but if not, you may visit Interfaces → (assign) and make a note of which physical interfaces, such as `fxp0`, correspond with the pfSense interfaces, such as WAN. Table 30.1, “Real Interface vs. Friendly Names” lists some interface names that you may encounter, depending on your configuration.

**Table 30.1. Real Interface vs. Friendly Names**

| Real/Physical Name | Friendly Name |
|---|---|
| enc0 | IPsec, encrypted traffic |
| ovpnc0 ... ovpnc<*x*>, ovpns0 ... ovpns<*x*> | OpenVPN, encrypted traffic (Clients, Servers) |
| pppoe0 ... pppoe<*x*>, poes0 ... poes<*x*> | PPPoE WAN, PPPoE Server |
| pptp0 ... pptp<*x*>, pptps0 ... pptps<*x*> | PPTP WAN, PPTP Server |
| l2tp0 ... l2tp<*x*>, l2tps0 ... l2tps<*x*> | L2TP WAN, L2TP Server |
| lo0 | Loopback Interface |
| pfsync0 | pfsync interface — used internally |
| pflog0 | pf logging — used internally |

When selecting an interface, you will typically want to start with where the traffic flows into pfSense. For example, if you are having trouble connecting to a port forward from outside your network, start with the WAN interface since that is where the traffic originates. Alternately, if you have a client PC which 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.

# Limiting capture volume

When capturing packets, it is important to limit the volume of packets captured, but still ensure all relevant traffic for the problem being troubleshooted is captured. On most networks, when capturing without filtering the traffic captured, even with captures from short time frames, you end up with huge amounts of data to dig through to find the problem. You can filter post-capture by using display filters in Wireshark, but filtering appropriately at the time of capture is preferable to keep the capture file size down. Filters are discussed later in this chapter.

# Packet Captures from the WebGUI

The WebGUI offers an easy-to-use front end to **tcpdump** that will let you get packet captures which can then be viewed or downloaded for deeper analysis in Wireshark. Because of its simplicity, it can only offer a few limited options for filtering desired traffic, which may complicate the task depending on the traffic level on your network and filtering needs. That said, for many people it is enough and gets the job done. If you feel limited by the options available, feel free to skip down to the next section on using **tcpdump** directly.

## Getting a Packet Capture

First, browse to Diagnostics → Packet Capture to start the process. From there, choose the Interface on which you would like to capture traffic. 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.

In previous versions of pfSense, the selected Interface was always placed into promiscuous mode when capturing traffic. Certain NICs do not handle promiscuous mode very well, so the default was changed. The Promiscuous checkbox is present to force the option if it is needed. In promiscuous mode, a capture will show all traffic arriving on the NIC no matter what the destination. Without promiscuous mode, only traffic destined for the host or broadcast will be captured.

The Address Family selection allows you to limit the capture to only IPv4 or only IPv6 traffic. This is only useful if you do not filter by IP address.

The Protocol drop-down lists a few common protocols such as TCP, UDP, ICMP, ICMP6, CARP and others. If you wish to limit the capture to one of these protocols, select it from the list. If you make

an invalid combination (e.g.IPv4 only and you choose ICMP6), the GUI will reject the input when you attempt to start the capture.

If you would like to filter traffic going to or from a specific host or CIDR-masked subnet, enter the IP address or subnet in the Host Address field.

The Port may be limited if you are capturing TCP or UDP traffic.

You can adjust the Packet Length to be captured if desired. Usually you will want the full packet, 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.

The Count box determines how many packets to capture before stopping. If you did not limit the capture in any way, bear in mind that this may be pretty "noisy" and you may need to increase this much larger than the default of **`100`**.

The Level of Detail option only affects the output as shown when the capture is finished. It does not change the level of detail in the capture file if you choose to download it when completed. Select how much detail you want in the GUI capture view.

It is not generally recommended to check Reverse DNS Lookups when performing a capture as it will delay the output as reverse DNS is performed. Also it is commonly easier to troubleshoot when viewing IP addresses instead of hostnames, and reverse DNS can sometimes be inaccurate. This can be useful on occasion though.

Press Start to begin capturing data. The screen will display "Packet Capture is running" across the bottom, indicating the capture is in process. Press Stop to end the capture and view the output. If you specified a maximum packet count it will stop automatically when that count is reached, or you can click Stop to end it at any time.

## 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 the section called “Viewing Packet Capture File” later in this chapter.

If you return to the packet capture page after a capture has been completed, you will see a View Capture button that will display the packets from the last capture run. You can 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 are shown in standard **tcpdump** style.

# Using tcpdump from the command line

**tcpdump** is the 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 by SSH. It is an exceptionally powerful tool, but that also makes it daunting to the uninitiated user. The **tcpdump** binary in FreeBSD 7.2 supports 36 different command line flags, limitless possibilities with filter expressions, and its man page, providing only a brief overview of all its options, is nearly 30 printed 8.5x11" pages long. After learning to use it, you must also know how to interpret the data it provides, 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 you 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 30.2. Commonly used tcpdump flags**

| Flag | Description |
|---|---|
| `-i <interface>` | Listen on `<interface>`, .e.g. `-i fxp0` |
| `-n` | Do not resolve IPs using reverse DNS. |
| `-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` | Don't put the interface in promiscuous mode. |
| `-v` | Verbose |
| `-e` | Print link-layer header on each line. Shows the source and destination MAC address, and VLAN tag information for tagged traffic. |

## -i flag

The `-i` flag specifies the interface on which **tcpdump** will listen. You use FreeBSD's interface name here, such as `fxp0`, `em0`, `rl0`, etc.

## -n flag

Do not resolve IPs 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. You may wish to disable this to avoid adding load to your DNS servers. I prefer to always use `-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 environments I am familiar with where I know the PTR records will provide the actual host names of the devices, I may run captures without `-n` to show the hostnames.

Another reason to use `-n`, though you should never capture in any environment where this is remotely a concern, is if you want to be "sneaky." One means of detecting packet capturing is looking for spikes and patterns in DNS PTR lookups.

## -w flag

**tcpdump** allows you to save capture files 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 [http://www.wireshark.org] or another graphical network protocol analyzer and reviewed there. When saving to a file using `-w`, the frames will not be displayed in your terminal as they otherwise are. (See the section called “Using Wireshark with pfSense” about using Wireshark with pfSense.)

## -s flag

By default, when capturing to a file, **tcpdump** will only save the first 64 bytes of each frame. 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, you can tell tcpdump how much of the frame to capture, in bytes. This is called the snap length.

**Table 30.3. 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 |

You will usually want to use `-s 0` when capturing to a file for analysis on another system. The only exception to this is scenarios where you need to capture a significant amount of traffic over a longer period of time. If you know the information you are seeking is in the header, you can save only the default 64 bytes of each frame and get the information you need, while significantly reducing the size of the resulting capture file.

## -c flag

You can instruct tcpdump to capture a certain number of frames and then exit by using 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 your network interface into promiscuous mode. When not running in promiscuous mode, your 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. In a switched network, this generally has little impact on your capture. In networks where the device you are capturing from is connected to a hub, using `-p` can significantly limit noise in your capture when the only traffic of interest is that to and from the system from which you are capturing.

## -v flag

The `-v` flag controls the detail, or verbosity, of the output. Using more "v" options yields more detail, so you can 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 em0 -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, 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(11
23:18:15.830851 IP 10.0.64.210.22 > 10.0.64.15.1395: P 116:232(116) ack 1 win 6
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
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 you see the link layer information included. Note the source and destination MAC addresses in addition to the source and destination IP addresses.

```
# tcpdump -ni em0 -e -c 5
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, 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),
23:30:05.915110 00:0c:29:0b:c3:ed > 00:13:d4:f7:73:d2, ethertype IPv4 (0x0800),
23:30:05.915396 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800),
23:30:05.973359 00:00:5e:00:01:04 > 01:00:5e:00:00:12, ethertype IPv4 (0x0800),
23:30:06.065200 00:13:d4:f7:73:d2 > 00:0c:29:0b:c3:ed, ethertype IPv4 (0x0800),
5 packets captured
```

# tcpdump Filters

On most firewalls, **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 allow you to limit the traffic displayed or captured to only what you are interested in seeing.

## 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 you can use the following command.

```
# tcpdump -ni em0 host 192.168.1.100
```

That will capture all traffic to and from that host. If you only wish to capture traffic being initiated by that host, you can use the `src` directive.

```
# tcpdump -ni em0 src host 192.168.1.100
```

Similarly, you can also filter for traffic destined to that IP address by specifying `dst`.

```
# tcpdump -ni em0 dst host 192.168.1.100
```

## Network filters

Network filters let you narrow down your capture to a specific subnet using the `net` expression. Following `net`, you can 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 em0 net 192.168.1
```

The next command is an example that will capture traffic to or from any host with a 10.x.x.x IP address.

```
# tcpdump -ni em0 net 10
```

Those examples will capture all traffic to or from the specified network. You can also specify `src` or `dst` the same as with `host` filters to capture only traffic initiated by or destined to the specified network.

```
# tcpdump -ni em0 src net 10
```

It is also possible to specify a CIDR mask as an argument to `net`.

```
# tcpdump -ni em0 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 your capture. Or you may not care about the source or destination of traffic, and simply wish to capture

a certain type of traffic. In other cases you may want to filter out all traffic of a specific type to reduce noise.

### TCP and UDP port filters

To filter on TCP and UDP ports you 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 em0 tcp port 80
```

#### Capture all DNS traffic

Capture all DNS traffic (usually UDP, but some queries use TCP).

```
# tcpdump -ni em0 port 53
```

### Protocol filters

You can filter by specific protocols using the `proto` directive. Protocol can be specified using the IP protocol number or one of the names `icmp`, `igmp`, `igrp`, `pim`, `ah`, `esp`, `vrrp`, `udp`, or `tcp`. Specifying `vrrp` will also capture CARP traffic as the two use the same IP protocol number. One common usage of the `proto` directive is to filter for CARP traffic. Because the normal protocol names are reserved words, they must be escaped with one or two backslashes, depending on the shell. The shell available in pfSense requires two backslashes to escape these protocol names. If you receive a syntax error, check that the protocol name is properly escaped. The following capture will show all CARP and VRRP traffic on the `em0` interface, which can be useful to ensure CARP traffic is being sent and received on the specified interface.

```
# tcpdump -ni em0 proto \\vrrp
```

## Negating a filter match

In addition to matching specific parameters, you can negate a filter match by specifying `not` in front of the filter expression. If you are troubleshooting something other than CARP and its multicast heartbeats are cluttering your capture output, you can exclude it as follows.

```
# tcpdump -ni em0 not proto \\vrrp
```

## Combining filters

You can combine any of the aforementioned filters using `and` or `or`. The following sections provide some examples.

### Display all HTTP traffic to and from a host

To display all HTTP traffic from the host 192.168.1.11, use the following command.

```
# tcpdump -ni em0 host 192.168.1.11 and tcp port 80
```

### Display all HTTP traffic to and from multiple hosts

To display all HTTP traffic from the hosts 192.168.1.11 and 192.168.1.15, use the following command.

```
# tcpdump -ni em0 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 en1 proto \\vrrp -c 2
tcpdump: syntax error
```

### Correct ordering

```
# tcpdump -ni en1 -c 2 proto \\vrrp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on en1, link-type EN10MB (Ethernet), capture size 96 bytes
18:58:51.312287 IP 10.0.64.3 > 224.0.0.18: VRRPv2, Advertisement, vrid 4, prio
18:58:52.322430 IP 10.0.64.3 > 224.0.0.18: VRRPv2, Advertisement, vrid 4, prio
2 packets captured
80 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 you 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 the section called “Additional References” at the end of this chapter for links to additional references on the subject.

# Practical Troubleshooting Examples

This section details an approach preferred by us 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 really happening on the network

## Port forward not working

You just added a port forward, and are trying to use it from a host on the Internet, but no dice. The troubleshooting steps outlined in the section called “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 you need to make sure the traffic is getting to your WAN interface. Start a **tcpdump** session on your WAN interface, and watch for the traffic.

```
# tcpdump -ni vlan0 tcp port 5900
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vlan0, 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(
```

In this case, we see a packet come 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 us. If the port forward is working you will see a SYN ACK packet 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 fxp0 tcp port 5900
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on fxp0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
11:14:38.339926 IP 172.17.11.9.2302 > 192.168.30.5.5900: S 1481321921:148132192
```

Looking at the internal traffic, we see that the connection did leave the inside interface, and the local IP address was translated correctly. If this local address matches what you expected, then both the port forward and the firewall rule are working properly, and connectivity to the local PC should be confirmed by other means. If you saw no output at all, then there is a problem with the firewall rule or the port forward may have been incorrectly defined. For this example, I had unplugged the PC.

## IPsec tunnel will not connect

Because **tcpdump** has some awareness of the protocols being used, it 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 may be more helpful in some cases, but this can confirm what is actually being seen by the router. For encrypted traffic such as IPsec, packet capturing of the traffic is of less value as you cannot examine the payload of the captured packets without additional parameters, but it is helpful to determine if traffic from the remote end is reaching your firewall and which phases complete.

This first tunnel has an unreachable peer:

```
# tcpdump -ni vr0 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vr0, 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 vr0 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vr0, 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
```

Now Phase 1 is OK but there is a mismatch in the Phase 2 information. It will repeatedly attempt phase 2 traffic but you won't see any traffic on the tunnel.

```
# tcpdump -ni vr0 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vr0, 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
19:17:18.525474 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others
19:17:19.527962 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others
```

Finally, a fully working tunnel with two-way traffic after Phase 1 and Phase 2 have completed!

```
#  tcpdump -ni vr1 host 192.168.10.6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vr1, 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
21:50:11.812809 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 2/others
21:50:12.820191 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others
21:50:12.836478 IP 192.168.10.6.500 > 192.168.10.5.500: isakmp: phase 2/others
```

```
21:50:12.838499 IP 192.168.10.5.500 > 192.168.10.6.500: isakmp: phase 2/others
21:50:13.168425 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x1), le
21:50:13.171227 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x1), le
21:50:14.178820 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x2), le
21:50:14.181210 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x2), le
21:50:15.189349 IP 192.168.10.5 > 192.168.10.6: ESP(spi=0x09bf945f,seq=0x3), le
21:50:15.191756 IP 192.168.10.6 > 192.168.10.5: ESP(spi=0x0a6f9257,seq=0x3), le
```

## Traffic traversing an IPsec tunnel

With some extra settings to initialize the process, you can also view traffic traversing your IPsec tunnels. This can help determine if traffic is attempting to reach the far end by using the tunnel. In versions prior to the 1.2.3 release, before **tcpdump** will work on the IPsec interface you had to set two **sysctl** variables that control what is visible to **tcpdump**. If you are using 1.2.3 release or newer, **tcpdump** will work without any extra handling.

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.

```
#  sysctl -w net.enc.out.ipsec_bpf_mask=0x00000001
net.enc.out.ipsec_bpf_mask: 0000000000 -> 0x00000001
#  sysctl -w net.enc.in.ipsec_bpf_mask=0x00000001
net.enc.in.ipsec_bpf_mask: 0000000000 -> 0x00000001
#  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 96 byt
22:09:18.331506 (authentic,confidential): SPI 0x09bf945f:
                IP 10.0.20.1 > 10.0.30.1:
                ICMP echo request, id 14140, seq 0, length 64
22:09:18.334777 (authentic,confidential): SPI 0x0a6f9257:
                IP 192.168.10.6 > 192.168.10.5: IP 10.0.30.1 > 10.0.20.1:
                ICMP echo reply, id 14140, seq 0, length 64 (ipip-proto-4)
22:09:19.336613 (authentic,confidential): SPI 0x09bf945f:
                IP 10.0.20.1 > 10.0.30.1:
                ICMP echo request, id 14140, seq 1, length 64
22:09:19.339590 (authentic,confidential): SPI 0x0a6f9257:
                IP 192.168.10.6 > 192.168.10.5: IP 10.0.30.1 > 10.0.20.1:
                ICMP echo reply, id 14140, seq 1, length 64 (ipip-proto-4)
```

If traffic was not properly entering the tunnel, you would not see any output. If there is a firewall or internal routing issue on the far side, you may see traffic leaving but nothing returning.

## Troubleshooting Outbound NAT

For complex environments where Advanced Outbound NAT is needed, **tcpdump** can be of great assistance in troubleshooting your Outbound NAT configuration. One good capture to use is to look for traffic with private IP addresses on your WAN interface, as everything you see on your WAN should be NATed to a public IP. 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 your outbound NAT rules, providing information to help review your Outbound NAT configuration to find the problem.

```
# tcpdump -ni em0 net 10 or net 192.168 or net 172.16.0.0/12
```

# Using Wireshark with pfSense

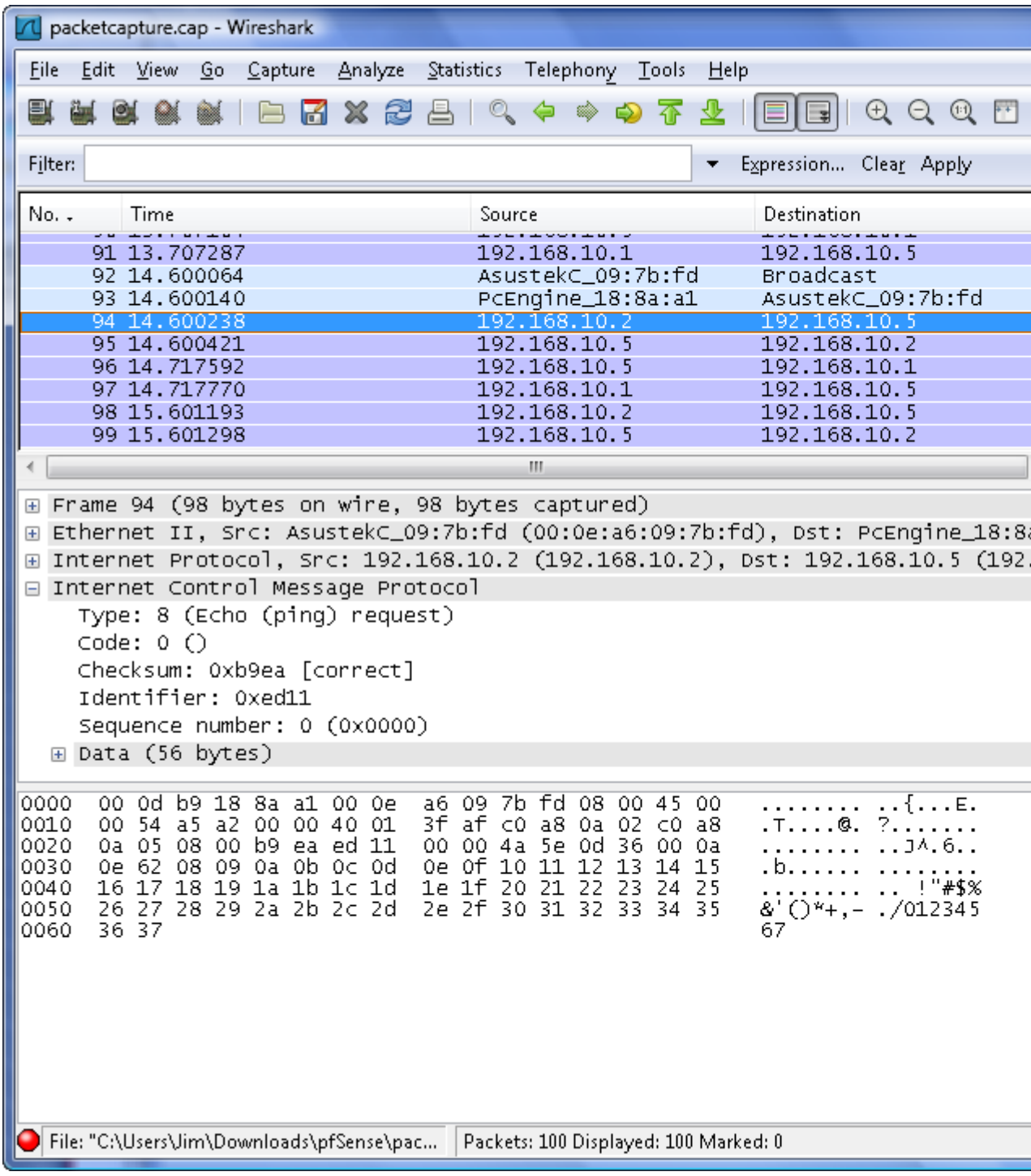
Wireshark, formerly known as Ethereal, is a GUI protocol analysis and packet capture tool that can be used to view and capture traffic much like **tcpdump**. It is Open Source software, freely available

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. You can also double click on any file with a `.pcap` extension in Windows and OS X with default settings after the Wireshark installation. You will see a screen similar to Figure 30.2, “Wireshark Capture View” in which the data from the capture file is displayed.

**Figure 30.2. Wireshark Capture View**



As seen in Figure 30.2, “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, you can filter the results using the Filter box on the toolbar. When you click on a packet, the lower frames will show the details of what was contained within the packet's 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, you will find a few options 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.

## Note

You will commonly see errors 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 you see in a packet capture is what is on the wire. Traffic sent out from the system where the capture is taken will have incorrect checksums where they are done in hardware, though traffic coming in from a remote system should always have correct checksums. You can turn off checksum offloading to ensure you are seeing traffic as the host is putting it on the wire, though usually this is something you simply ignore. Should you need to verify checksums, you will usually want to capture traffic from another system using a network tap or switch span port. Span ports can also be setup on bridges in pfSense, see the section called “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's 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.

### Figure 30.3. Wireshark RTP Analysis

| Src IP addr | Src port | Dest IP addr | Dest port | SSRC | Payload | Packets | Lost | Max De |
|---|---|---|---|---|---|---|---|---|
| 10 | 13114 | 192 | 2244 | 0x63B69143 | ITU-T G.711 PCMU | 2103 | 0 (0.0%) | |
| 192.1 | 2244 | 1 | 13114 | 0x6F1D173B | ITU-T G.711 PCMU | 1646 | 477 (22.5%) | |
| 10 | 11224 | 192 | 2268 | 0x2247C8D8 | ITU-T G.711 PCMU | 1321 | 0 (0.0%) | |
| 192.1 | 2268 | 1 | 11224 | 0x6C5B26A1 | ITU-T G.711 PCMU | 879 | 460 (34.4%) | |
| 10 | 17924 | 192 | 2242 | 0x393CBA89 | ITU-T G.711 PCMU | 480 | 0 (0.0%) | |
| 192.1 | 2242 | 1 | 17924 | 0x6177246E | ITU-T G.711 PCMU | 133 | 366 (73.3%) | |

# Remote Realtime Capture

From a UNIX host that has Wireshark available, you can run a realtime remote capture by redirecting the output from an SSH session. This has been tested and known to work on FreeBSD and Ubuntu.

In order to use this technique, SSH must be enabled on the pfSense system and you will need to use an SSH key (see the section called “Secure Shell (SSH)”). The key must first be loaded into **ssh-agent** or generated without a passphrase because the redirection will not allow you to enter a password. Using **ssh-agent** is highly recommended, as any key without a passphrase is very insecure.

Before you attempt this technique, be sure that you can connect to your pfSense router using an SSH key without needing to type the passphrase. The first time you connect, you will be prompted to save the host key, so that must also be done before you try to start wireshark. You may start **ssh-agent** from a terminal window or shell like so:

```
# eval `ssh-agent`
Agent pid 29047
# ssh-add
Enter passphrase for /home/jim/.ssh/id_rsa:
Identity added: /home/jim/.ssh/id_rsa (/home/jim/.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 1.2.3-pfSense on exco-rtr ***
[...]
```

After you have confirmed that the SSH connection works, start the remote capture as follows:

```
# wireshark -k -i <(ssh root@192.168.1.1 tcpdump -i vr0 -U -w - not tcp port 22
```

Where the IP address part is the address of your pfSense system. The "`not tcp port 22`" part will exclude the traffic from your SSH session, which will otherwise clog the capture output. The above is written in "bash-style" syntax, but may work with other shells. You can adjust the **tcpdump** arguments for the interface, and add additional expressions, but 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 [http://wiki.wireshark.org/CaptureSetup/Pipes] page on the Wireshark wiki for other related techniques.

# Plain Text Protocol Debugging with tcpflow

**tcpflow** is another package similar to **tcpdump** which will let you view the text contents of packets in realtime instead of the packet headers and other transport information. **tcpflow** uses similar syntax to **tcpdump**, with one notable exception: By default it writes the packet text to files instead of the console. To watch output on the console, use the `-c` option.

While not available on a stock pfSense installation, **tcpflow** may be added from the command line by installing the FreeBSD package. It is a small package with no dependencies, so installing it should not harm the system. To install **tcpflow** on pfSense, run the following command from a pfSense shell:

```
# pkg_add -r tcpflow
# rehash
```

If you were having trouble with an FTP connection from a LAN, you could monitor the control channel on the WAN side like so:

```
# tcpflow -i em0 -c host 172.17.11.9 and port 21
tcpflow[13899]: listening on vlan0
```

```
172.017.011.009.00021-010.000.073.005.23747: 220 Welcome to ExampleCo web FTP s
010.000.073.005.23747-172.017.011.009.00021: USER fieldtech
172.017.011.009.00021-010.000.073.005.23747: 331 Please specify the password.
010.000.073.005.23747-172.017.011.009.00021: PASS abc123
172.017.011.009.00021-010.000.073.005.23747: 230 Login successful.
010.000.073.005.23747-172.017.011.009.00021: PORT 10,0,73,5,194,240
172.017.011.009.00021-010.000.073.005.23747: 200 PORT command successful. Consi
010.000.073.005.23747-172.017.011.009.00021: NLST
172.017.011.009.00021-010.000.073.005.23747: 150 Here comes the directory listi
172.017.011.009.00021-010.000.073.005.23747: 226 Directory send OK.
```

As you can see from this output, it is easy to monitor the flow of plain text control protocols like FTP. You can see commands and output going in both directions, and most importantly you can see that the FTP proxy did its job and translated the PORT command to use the WAN IP address of pfSense instead, allowing active mode to work properly. If you instead saw the LAN IP address listed in the PORT command, you would know to check the FTP proxy settings or switch to PASV mode on the client.

Having **tcpflow** around has been very handy in my experience, and it makes a good complement to **tcpdump** when you want to focus on the contents of the packets rather than their structure.

# Additional References

This capture only scratches the surface of the possibilities with packet captures. Here are some additional resources for those interested in more in-depth knowledge. Packet capturing is a very powerful means of troubleshooting network connectivity issues, and you will find your troubleshooting skills greatly improved if you learn the possibilities in more depth.

*Computer Networking: Internet Protocols in Action* [http://www.amazon.com/gp/product/0471661864?ie=UTF8&tag=pfsense-20&linkCode=as2&camp=1789&creative=9325&creativeASIN=0471661864] by Jeanna Matthews

*Tcpdump Filters* [http://www.whitehats.ca/main/members/Malik/malik_tcpdump_filters/malik_tcpdump_filters.html] by Jamie French

*Tcpdump Advanced Filters* [http://acs.lbl.gov/~jason/tcpdump_advanced_filters.txt] by Sebastien Wains

*Tcpdump Filters* [http://www.cs.ucr.edu/~marios/ethereal-tcpdump.pdf] by Marios Iliofotou

FreeBSD Man Page for **tcpdump** [http://www.freebsd.org/cgi/man.cgi?query=tcpdump&apropos=0&sektion=0&manpath=FreeBSD+7.2-RELEASE&format=html]

# Appendix A. Menu Guide

This guide to the standard menu choices available in pfSense should 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 you may have to check all of them to locate the menu options for any installed packages. Typically, packages install under the Services menu but there are plenty of them that occupy other menus as well.

## System

The System menu contains choices for the system itself, general and advanced options, firmware updates, add-on packages, and static routes.

| | |
|---|---|
| Advanced | Advanced system settings for the firewall, hardware, SSH, SSL certificates, and many others. See the section called “Advanced Configuration Options”. |
| Cert Manager | Manage Certificate Authorities, Certificates, and Certificate Revocation Lists (x.509). See Chapter 8, *Certificate Management*. |
| Firmware | Upgrade or change the system firmware version. (e.g. update from pfSense 1.2.2 to 1.2.3). See the section called “Upgrading using the WebGUI”. |
| General Setup | General system settings such as hostname, domain, DNS servers, etc. See the section called “General Configuration Options”. |
| High Avail. Sync | Formerly known as "CARP Settings", this section controls how pfSense nodes in a High Availability (HA) cluster synchronize states and configuration. See Chapter 25, *Firewall Redundancy / High Availability*. |
| Logout | Logs out of the GUI, returning the user back to the login screen. See Chapter 7, *User Management and Authentication*. |
| Packages | Additional software add-ons for pfSense to expand its functionality. See Chapter 28, *Packages*. |
| Routing | This is where you define gateways, static routes, and gateway groups for multi-WAN. See Chapter 12, *Routing*. |
| Setup wizard | The Setup Wizard guides you through the process of performing the basic initial setup. See the section called “Setup Wizard”. |
| User Manager | Manage users, groups, and authentication servers (RADIUS or LDAP) for GUI access, VPN access, etc. See Chapter 7, *User Management and Authentication*. |

## Interfaces

The Interfaces menu has items for assigning interfaces, and an item for each assigned interface. WAN and LAN will always appear, while others appear as OPTx or their chosen custom name.

| | |
|---|---|
| (assign) | Assign interfaces to logical roles (e.g. LAN, WAN, OPT), and create/maintain VLANs and other types of virtual interfaces. See the section called “Assign interfaces”, Chapter 6, *Interface Types and Configuration*, and Chapter 14, *Virtual LANs (VLANs)*. |
| WAN | Configure the WAN interface. See the section called “Interface Configuration Basics”. |
| LAN | Configure the LAN interface. See the section called “Interface Configuration Basics”. |

| | |
|---|---|
| OPTx | Configure any additional optional interfaces. See the section called “Interface Configuration Basics”. |

# Firewall

The Firewall menu items are for configuring various parts of the firewall rules, NAT rules, and their supporting structure.

| | |
|---|---|
| Aliases | Lets you manage collections of IP addresses, networks, or ports to simplify rule creation and management. See the section called “Aliases”. |
| NAT | Maintain NAT rules that control port forwards, 1:1 NAT, and outbound NAT behavior. See Chapter 11, *Network Address Translation*. |
| Rules | Configure firewall rules. There should be one tab on this screen for each configured interface. See the section called “Introduction to the Firewall Rules screen”. |
| Schedules | Setup time-based rule schedules. See the section called “Time Based Rules”. |
| Traffic Shaper | Configure traffic shaping/Quality of Service (QoS) settings. See Chapter 21, *Traffic Shaper*. |
| Virtual IPs | Configure Virtual IP addresses to let pfSense handle traffic for more than one IP address per interface, typically for NAT rules or CARP failover. See the section called “Virtual IPs”. |

# Services

The Services menu contains items which allow you to control various services provided by daemons running on pfSense. See Chapter 26, *Services*.

| | |
|---|---|
| Captive portal | Controls the Captive Portal service which allows you to direct users to a web page first for authentication before permitting Internet access. See Chapter 24, *Captive Portal*. |
| DHCP relay | Configures the DHCP relay service which will proxy DHCP requests from one network segment to another. See the section called “DHCP & DHCPv6 Relay”. |
| DHCP server | Configures the DHCP service which provides automatic IP address configuration for clients on Internal interfaces. See the section called “IPv4 DHCP Server”. |
| DHCPv6 Relay | Configures the DHCP relay service for IPv6 which will proxy DHCP requests from one network segment to another. See the section called “DHCP & DHCPv6 Relay” |
| DHCPv6 Server/RA | Configures the DHCP service for IPv6 and Router Advertisements which provide automatic IPv6 address configuration for clients on Internal interfaces. See the section called “IPv6 DHCP Server and Router Advertisements” |
| DNS forwarder | Configures pfSense's built-in caching DNS resolver. See the section called “DNS Forwarder”. |
| Dynamic DNS | Configures Dynamic DNS services (dyndns) which will update a remote system when this pfSense router's WAN IP address has changed. See the section called “Dynamic DNS”. |

| | |
|---|---|
| Load Balancer | Configures the Load Balancer, which balances incoming connections across multiple servers. See Chapter 22, *Server Load Balancing*. |
| NTP | Configure the Network Time Protocol server daemon. See the section called “NTPD”. |
| PPPoE Server | Configure the PPPoE server which allow pfSense to accept and authenticate connections from PPPoE clients. See the section called “PPPoE Server”. |
| SNMP | Configures the Simple Network Management Protocol (SNMP) daemon to allow network-based collection of statistics from this router. See the section called “SNMP”. |
| UPnP & NAT-PMP | Configure the Universal Plug and Play (UPnP) service which can automatically configure NAT and firewall rules for devices which support the UPnP standard. See the section called “UPnP & NAT-PMP”. |
| Wake on LAN | Configure Wake on LAN services which allow you to remotely wake up client PCs reachable from the pfSense system. See the section called “Wake on LAN”. |

# VPN

The VPN menu contains items pertaining to Virtual Private Networks (VPNs), including IPsec, OpenVPN and PPTP. See Chapter 16, *Virtual Private Networks*.

| | |
|---|---|
| IPsec | Configure IPsec VPN tunnels, mobile IPsec options and users, and certificates. See Chapter 17, *IPsec*. |
| L2TP | Configure L2TP services and users. See Chapter 19, *PPTP VPN*. |
| OpenVPN | Configure OpenVPN servers and clients, as well as client-specific configuration. See Chapter 18, *OpenVPN*. |
| PPTP | Configure PPTP services and users, or relay. See Chapter 19, *PPTP VPN*. |

# Status

The Status menu allows you to check the status of various system components and services, as well as view logs.

| | |
|---|---|
| Captive Portal | When Captive Portal is enabled, you can view user status here. See Chapter 24, *Captive Portal*. |
| CARP (failover) | View the status of CARP IP addresses on this system. Will show MASTER/BACKUP status. See the section called “Check CARP status”. |
| Dashboard | A shortcut back to the main page of the pfSense router that displays general system information. See the section called “Dashboard”. |
| DHCP leases | View a list of all DHCP IPv4 leases assigned by this router. You can also delete offline leases, send Wake on LAN requests to offline systems, or create static leases from current entries. See the section called “Leases”. |
| DHCPv6 leases | View a list of all DHCP IPv6 leases assigned by this router. You can also delete offline leases, send Wake on LAN requests to offline systems, or create static leases from current entries. See the section called “Leases”. |
| Filter Reload | Shows the status of any filter reload requests that are (or were) pending. The filter is reloaded whenever changes are applied. If no changes have been made, this screen should simply report that an update has been completed. |

| | |
|---|---|
| Gateways | Shows the status of gateways, and gateway groups for multi-WAN. See Chapter 12, *Routing*. |
| Interfaces | Lets you view the hardware status for network interfaces, equivalent to using **ifconfig** on the console. See the section called “Interface Status”. |
| IPsec | Views the status of any configured IPsec tunnels. See Chapter 17, *IPsec*. |
| Load Balancer | Views the status of the Load Balancer pools. For gateway load balancing, see the section called “Testing Failover”. For server load balancing see the section called “Viewing load balancer status”. |
| NTP | Views the status of the Network Time Protocol server daemon. See the section called “NTPD”. |
| OpenVPN | Views the status of any configured OpenVPN instances. See the section called “Checking the Status of OpenVPN Clients and Servers”. |
| Package logs | View logs from certain supported packages. |
| Queues | View the status of the traffic shaping queues. See the section called “Monitoring the Queues”. |
| RRD Graphs | View graphed data for system statistics such as bandwidth used, CPU usage, firewall states, etc. See the section called “RRD Graphs”. |
| Services | Monitor the status of system and package services/daemons. See the section called “Service Status”. |
| System logs | View logs from the system and system services such as the firewall, DHCP, VPNs, etc. See the section called “System Logs”. |
| Traffic graph | View a dynamic SVG-based realtime traffic graph for an interface. See the section called “Traffic Graphs”. |
| UPnP & NAT-PMP | View a list of any currently active UPnP port forwards. See the section called “UPnP & NAT-PMP”. |
| Wireless | View a list of any currently available wireless networks in range. See the section called “Showing available wireless networks and signal strength”. |

# Diagnostics

Items under the Diagnostics menu perform various diagnostic and administrative tasks.

| | |
|---|---|
| ARP Tables | View a list of systems as seen locally by the router. The list includes an IP address, MAC address, Hostname, and the Interface where the system was seen. |
| Authentication | Tests authentication to a defined RADIUS or LDAP server. See the section called “Troubleshooting”. |
| Backup/Restore | Backup and restore configuration files. See the section called “Making Backups in the WebGUI”, the section called “Restoring with the WebGUI”, and the section called “Restoring from the Config History”. |
| Command Prompt | Execute shell commands or PHP code, and upload/download files to the pfSense system. **Use with caution**. |
| DNS Lookup | Executes a DNS lookup to resolve hostnames for diagnostic purposes, and to test connectivity to your DNS servers. See the section called “Testing DNS” |

| | |
|---|---|
| Edit File | Edit a file on the pfSense system. |
| 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. |
| Halt system | Shut down the router and turn off the power where possible. |
| Limiter Info | Shows the status of any Limiters and the traffic flowing inside them. See Chapter 21, *Traffic Shaper*. |
| NDP Table | View a list of local IPv6 systems as seen by the router. The list includes an IPv6 address, MAC address, hostname (if known to the firewall), and the interface. |
| NanoBSD | Only visible on the NanoBSD (embedded) platform. Allows cloning of the working slice over to the alternate slice, and choose which one should be used to boot the router. See the section called “NanoBSD-Specific Configuration” |
| Packet Capture | Perform a packet capture to inspect traffic, and then view or download the results. See the section called “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 the section called “pfInfo” |
| pfTop | Displays a list of the top active connections by a selectable metric such as bytes, rate, age, etc. See the section called “Viewing with pftop” |
| Ping | Send ICMP echo requests to a given IP address, sent via a chosen interface. |
| Reboot system | Reboot the pfSense router. Depending on the hardware, this could take several minutes. |
| Routes | Shows the contents of the system's routing table. See the section called “Viewing Routes”. |
| SMART Status | Displays diagnostic information about IDE drives, if supported by the hardware. Can also run hard drive tests. See the section called “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 | View the currently active firewall states. See the section called “Viewing in the WebGUI”. |
| States Summary | Displays information about the state table, to see activity summarized by IP address. See the section called “States Summary” |
| System Activity | Shows memory usage and a list of active processes and system threads on the firewall, the output is from **top -SH**. See the section called “System Activity (Top)” |
| Tables | Allows you to view and edit the contents of various system tables and aliases. See the section called “Viewing the Contents of Tables” |
| Traceroute | Trace the route taken by packets between the pfSense router and a remote system. See the section called “Using traceroute”. |

# Index

## Symbols



## A



## B



## C

## D



## E



## F



## G

## H



## I

## K



## L



## M

## N



## O

## S



## T

## U



## V



## W

## X