

TALUG Meeting Notes

April 26, 2008

pfSense: Presented by Neal Dudley

We got started about 20 minutes late due to some hardware problems. Neal had brought his Apple laptop, which uses the (non)standard Apple video connection, and he hadn't remembered to bring his adapter. Also Neal had not realized that the Solaris machines were not standard architecture, and therefore would not be able to boot pfSense, or any other distribution.

What could have been a disaster was quickly saved by hardware donations from the audience. Eric donated the use of his laptop to boot the pfSense live CD, Andrew donated his laptop's network card, and Matt donated his laptop to act as client computer running the web interface. What better place to get people to quickly throw a bunch of hardware together to build a firewalled network than a Linux meeting? Plus, it showed the excellent hardware support of [pfSense](#), being able to run on a hodge podge system.

Physical Setup

The actual setup used was one laptop running pfSense, and one client laptop running Fedora. The pfSense laptop had two network cards, one connected to the Internet and the other connected to the local network. We used a cross-over cable to directly hook the pfSense laptop up to the client laptop.

Overview

[pfSense](#) is an inexpensive solution for setting up a firewall, and monitoring bandwidth usage. PfSense started as a fork of [m0n0wall](#), but focuses more on PC installations than on embedded hardware. Additionally, pfSense has a lot of additional features over m0n0wall including vlans, traffic shaping etc. PfSense also has good ppoe support that actually works as advertised. One of the cool things that pfSense can do is set up *CARP* (Common Address Redundancy Protocol) with two pfSense boxes. In this situation, if one pfSense box goes down for any reason, the remaining box will take up all the extra traffic (transparent to the user). PfSense is based on FreeBSD, and has really good hardware support. Without any help, it recognized all 3 NICs on Eric's laptop (2 wired and 1 wireless).

The hardware requirements for pfSense are really quite modest, 100 Mhz cpu, and 128 megs of ram.

Boot-Up and CLI Interface

During the boot-up sequence of pfSense, we briefly discussed vlans and "router on a stick." PfSense gave a number of prompts on boot-up to choose things like the NIC facing the local network, and the NIC facing the Internet. All in all, the boot-up options were rather straight forward and easy to figure out.

The login counsel of pfSense displays a simple menu, offering a number of options (see below).

```
pfSense console setup
*****
0) Logout (SSH only)
1) Assign Interfaces
2) Set LAN IP address
3) Reset webConfigurator password
4) Reset to factory defaults
5) Reboot system
6) Halt system
7) Ping host
8) Shell
9) PFtop
10) Filter Logs
11) Restart webConfigurator
12) pfSense PHP shell
13) Upgrade from console
99) Install pfSense to a hard drive/memory drive, etc.
```

As an example of the command line interface, Neal showed PFtop, which gave an overview of the traffic going through the firewall. We went through the other options one by one and Neal explained each one.

PfSense provides command line access from either the local box, or via ssh from the lan, or ssh via the wan (disabled by default). However, Neal pointed out that the majority of the configuration happens via the web interface.

Web Interface

After our brief foray into the command line side of pfSense, we switched the projector over to Matt's laptop to view the web gui. We logged in via the web interface and started going through the initial configuration.

It was pointed out that Eric's laptop had a lot more power than what would typically be found on a firewall box. Therefore loading the initial configuration of pfSense on slower hardware could take a long time. So depending on your hardware, expect to wait a while.

The first things you have to set up are a hostname, domain, dns server, and a time server. depending on the ISP, you may need to spoof the mac address (easy in the web gui).

The next steps were to configure the subnet, and set up a username and password for the gui account. It was noted that the admin and web-admin accounts are completely separate. This allows for granular levels of accounts. Certain users can be set up to use certain parts of pfSense, without allowing complete access.

Once the basic setup was out of the way, we arrived at the system overview page. This page shows some basic hardware stats, including cpu, memory, and disk usage. Eric's laptop was overpowered, so we were using almost nothing.

We then entered the web configuration settings, and talked about the following:

- The web gui is theme-able, so it's appearance can be changed to suit your preference.
- DHCP can be relayed to an external server.
- Configuration of wireless AP. Can run radius locally.
- The web gui has an anti lockout feature. Has options, lets you screw yourself if you really want to.
- PfSense tells you what it's doing as it is doing it, giving positive feedback. Example: There is a scroll bar across the top showing the current status, and what it is working on.
- Traffic shaping: Unfortunately, it was not supported on Andrew's 10 year old 3com network card.

- Scheduling: for example, if you only want a lab to be open from 8 to 5, block certain services at certain times. Another Example: don't allow anything to touch your WiFi while you are not at home.
- Policy based routing: instead of referring to traffic from a certain interface, define traffic by type and then implement certain rules for that type of traffic. If you add NAT rules, it will prompt you to add them automatically to the firewall.
- You can block traffic for different types of operating systems: Windows, BSD, NetBSD, Linux. For example, Vista tries to go around the firewall for IPV6. Can just block anything from a windows box on a certain port.
- Set the maximum number of connections. Advanced options, set number of tries per unit time for a certain service. Example: ssh connection attempts per second.
- Logging.
- IPV6.
- Can act as time server (open NTP).
- Diagnostic packet capture.
- Firewall services: we can designate port ranges by services. Lots of default port/services are shown to make things easy.
- State modulation, can have pfSense "mangle" packet numbers so that you can't guess what the next one is.
- VPN (openVPN) if you want to set up a VPN at home but are stuck with NAT (uses regular UDP port), you can set up an open VPN server. PfSense can run the VPN server internally.
- Command prompt through the web-gui. Works decently, but kind of hokey. Example: we ran `uname -a` and it returned the OS information. It also has interface for file to upload and file to download. We download the `pf.conf` file to see how much work we saved using the web gui.
- Backup and restore feature so that you don't have to worry losing your settings and having to do it all over again. Backups are defined in xml.

Next we went back to look at a few of the configuration options in more detail. We started with firewall rules, and defined a new rule to block ICMP echos, so that pfSense wouldn't respond to a ping.

Neal asked for people to ping the server, and we went to watch the command line logging (PFtop) as a large number of ping requests from a variety of IP's were logged.

Then we went back to the web-gui and looked through the system logs. It has a nice log format that makes it easy to see what is going all.

We then set up external ssh access.

Note: 5 Minute Break

Neal noted that this was like a national lampoon's pfSense presentation: Hardware problems, bathroom break, and internal sabotage (ssh access). Ssh access was quickly disabled.

We then looked at the web gui's traffic graphs (svg's, so they won't work in Internet Explorer). The traffic graphs are updated in real time, and can quickly show what is going on on the network. There are picture of the traffic graphs on the [talug website](#).

Installation and Hardware

- PfSense comes in two different versions: as a livecd, or as an embedded version.
- Neal recommends an ide/sata → flash connector so that you don't have to worry about hard drive failures.
- Went over packages / whats available / how to install.

Documentation

[PfSense documentation](#): it doesn't suck as much as you would think! They have video tutorials to show how to set things up, it is actually pretty cool.