Firewalls in FreeBSD and OpenBSD

What is firewall?

- **firewall** is a method of protecting hosts and networks connected to other hosts and networks against attacks from the outside and from the inside.
- a **firewall** is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.
- A **firewall** typically establishes a barrier between a trusted, secure internal network and another outside network, such as the Internet, that is assumed not to be secure or trusted

How many firewall run in FreeBSD??

- Ipfw
- pf
- IPFILTER

What is ipfw?

- HISTORY:

The **ipfw** utility first appeared in **FreeBSD 2.0**. **dummynet(4)** was introduced in FreeBSD 2.2.8. Stateful extensions were introduced in **FreeBSD 4.0**. ipfw2 was introduced in Summer 2002.

ipfw

- IPFW is a statefull firewall written for FreeBSD which supports both IPv4 and IPv6.

- kernel firewall filter rule processor and its integrated packet accounting facility, the **logging** facility, **NAT**, the **dummynet** traffic shaper, a **forward** facility , **bridge** facility.

Enabling IPFW in FreeBSD

- IPFW is included in the basic FreeBSD install as a kernel loadable module,(kldload) in rc.conf.

- statically compile IPFW support into a custom kernel:

IPFW Syntax

- ipfw add 110 deny log all from any to 127.0.0.0/8
- ipfw add 120 deny log all from any to 127.0.0.0/8
- #Start dynamic state filtering

  ipfw add 200 check-state
- #Filter out fragmented packets that do not have an offset of one which are automatically dropped

  ipfw add 210 deny all from any to an frag in via en0

IPFW Command

- To list all the running rules in sequence:

```
# ipfw list
```

- lists accounting information and the packet count for matched rules along with the rules themselves:

```
# ipfw -a list
```

- The syntax for flushing the chain is:

```
#ipfw flush
```

Enabling IPFW and DummyNet

- Load dummynet with:

```
#kldload dummynet
```

- To create a pipe

```
#ipfw add 100 pipe 1 ip from any to any
```

- This command will show all the parameters associated with the pipe:

```
#ipfw pipe 1 show
```

Enabling IPFW and DummyNet

- For destroy the pipe 1:

```
#ipfw pipe 1 delete
```

- destroy all the pipes generated:

```
#ipfw pipe flush
```

DummyNet(Bandwidth)

- Bandwidth Config:

Setting the bandwidth of the traffic between the hosts. The bandwidth can be any of bit/s , Kbit/s,Mbits/s, Byte/s. KByte/s , MByte/s. A bandwidth of zero results in no bandwidth limitation.

```
#ipfw pipe 1 config bw  100Kbit/s
```

This command limits the bandwidth of the pipe 1 to 100Kbit/s.

DummyNet(Queue Size)

- The queue size can also be set, which along with bandwidth influences the queueing delay. The queue size can be specified as number of slots, in Bytes or in KBytes.

```
#ipfw pipe 1 config queue 100KByte/s
```

- This command limits the queue size of the pipe 1 to 100KByte/s.

DummyNet(Delay)

- The prorogation delay of the pipes can also be controlled and can be set to any desired value a in milliseconds. The documentation states that the queueing delay is independent of the prorogation delay.

```
#ipfw pipe 1 config delay 100ms
```

- This command sets the desired propogation delay to 100ms.

DummyNet(Random Packet Loss)

- The packet loss in a network can also be simulated in the dummynet. The command plr X, where X is a floating point number between 0 and 1 which causes packets to be dropped at random simulates packet loss, where 0 is for no loss and 1 is for 100%packet loss.

```
#ipfw pipe 1 config plr 0.5
```

- This command drops packets randomly , sending almost half the number of packets across the network.

PF fireWall

- PF was originally designed as replacement for Darren Reed's IPFilter, from which it derives much of its rule syntax. IPFilter was removed from OpenBSD's CVS tree on 30 May 2001 due to OpenBSD developers' concerns with its license.
- he initial version of PF was written by Daniel Hartmeier. It appeared in OpenBSD 3.0, which was released on 1 December 2001.

Pf Features

- Network Address Translation (NAT) and Quality of Service (QoS) have been integrated into PF, QoS by importing the ALTQ queuing software and linking it with PF's configuration.
- s pfsync and CARP for failover and redundancy.
- authpf for session authentication.
- ftp-proxy to ease firewalling the difficult FTP protocol.
- PF's logging. PF's logging is configurable per rule within the pf.conf and logs are provided from PF by a pseudo-network interface called pflog

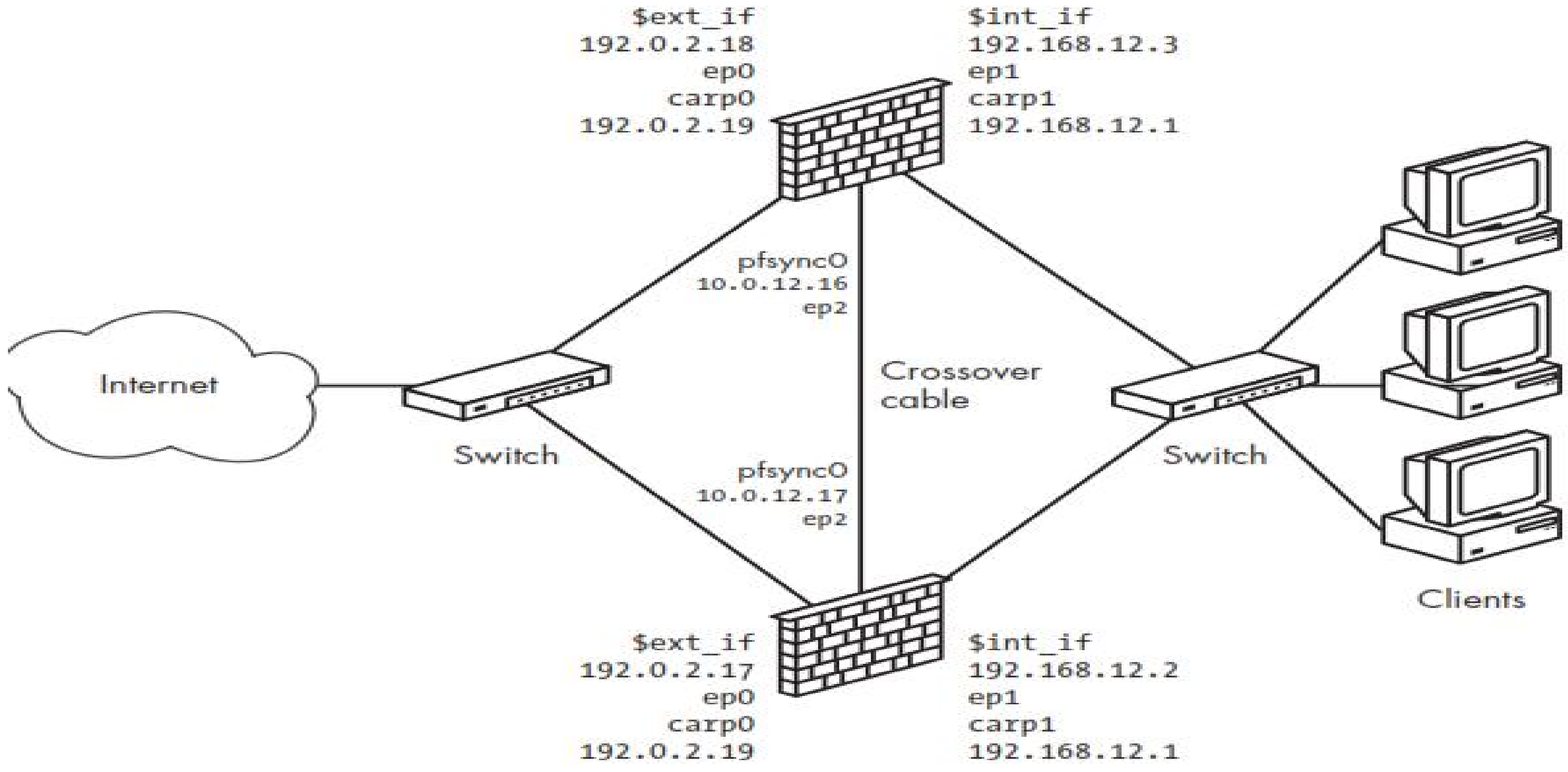
What is CARP?

- CARP is the Common Address Redundancy Protocol.

- Its primary purpose is to allow multiple hosts on the same network segment to share an IP address.

- CARP works by allowing a group of hosts on the same network segment to share an IP address. This group of hosts is referred to as a "redundancy group“.

What is CARP?


$ext_if
192.0.2.18
ep0
carp0
192.0.2.19
$int_if
192.168.12.3
ep1
carp1
192.168.12.1
pfsync0
10.0.12.16
ep2
Internet
Switch
Crossover
cable
Switch
pfsync0
10.0.12.17
ep2
Clients
$ext_if
192.0.2.17
ep0
carp0
192.0.2.19
$int_if
192.168.12.2
ep1
carp1
192.168.12.1

CARP Example

- Here is an example CARP configuration:

```
# sysctl net.inet.carp.allow=1
# echo 'net.inet.carp.allow=1' >> /etc/sysctl.conf
# ifconfig carp1 create
# ifconfig carp1 vhid 1 pass mekmitasdigoat carpdev em0 advskew 100 10.0.0.1 netmask 255.255.255.0
```

CARP Example

- Enables receipt of CARP packets (this is the default setting).
- Creates a carp(4) interface, carp1.
- Configures carp1 for virtual host #1, enables a password, sets em0 as the interface belonging to the group, and makes this host a backup due to the advskew of 100 (assuming of course that the master is set up with an advskew less than 100). The shared IP assigned to this group is 10.0.0.1/255.255.255.0.

Introduction to pfsync

- The pfsync(4) network interface exposes certain changes made to the pf(4) state table.
- By monitoring this device using tcpdump(8), state table changes can be observed in real time.
- In addition, the pfsync(4) interface can send these state change messages out on the network so that other nodes running PF can merge the changes into their own state tables.

pfsync Example

- Here is an example pfsync configuration:

```
# ifconfig pfsync0 syncdev em1 up
```

- This enables pfsync on the em1 interface. Outgoing updates will be multicast on the network allowing any other host running pfsync to receive them.

Combining CARP and pfsync For Failover

- By combining the features of **CARP** and **pfsync**, a group of two or more firewalls can be used to create a highly-available, fully redundant firewall cluster.
- CARP:

Handles the automatic failover of one firewall to another.

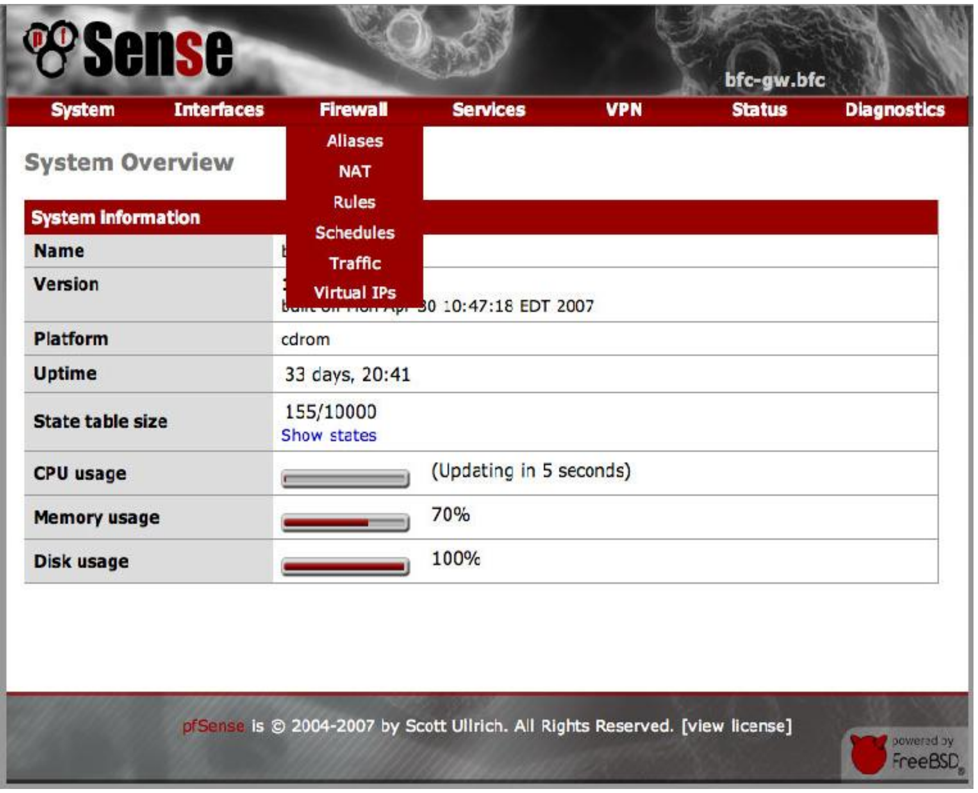
- pfsync:

Synchronizes the state table amongst all the firewalls. In the event of a failover, traffic can flow uninterrupted through the new master firewall.

pfSense

- pfSense is an open source firewall/router computer software distribution based on FreeBSD.
- it can be configured and upgraded through a web-based interface, and requires no knowledge of the underlying FreeBSD system to manage.
- The pfSense project started in 2004 as a fork of the m0n0wall project by Chris Buechler and Scott Ullrich.

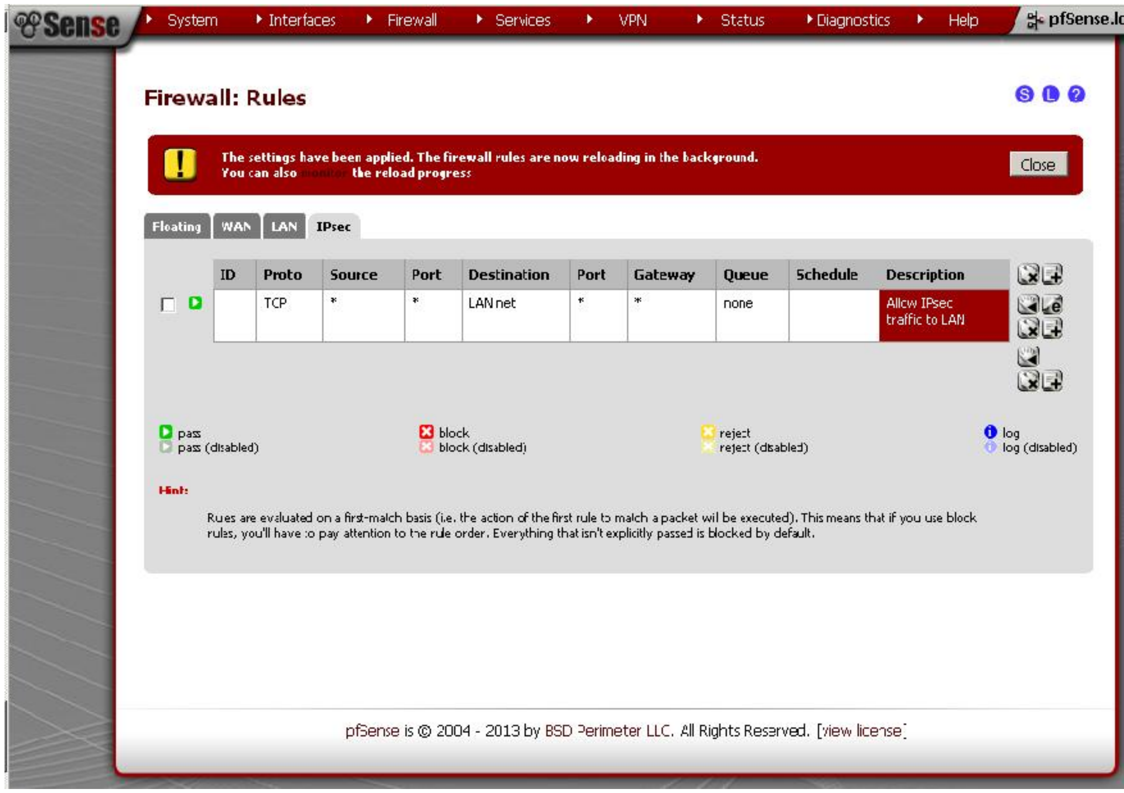
pfSense



System Overview

System Information	
Name	b
Version	1 built on Mon Apr 30 10:47:18 EDT 2007
Platform	cdrom
Uptime	33 days, 20:41
State table size	155/10000 Show states
CPU usage	(Updating in 5 seconds)
Memory usage	70%
Disk usage	100%

pfSense is © 2004-2007 by Scott Ullrich. All Rights Reserved. [view license]



Firewall: Rules

The settings have been applied. The firewall rules are now reloading in the background.
You can also monitor the reload progress

		ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
☐	▶		TCP	*	*	LAN net	*	*	none		Allow IPsec traffic to LAN

▶ pass
▷ pass (disabled)
✖ block
✖ block (disabled)
✖ reject
✖ reject (disabled)
ℹ log
ℹ log (disabled)

Hint:

Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

pfSense is © 2004 - 2013 by BSD Perimeter LLC. All Rights Reserved. [view license]

m0n0wall

- m0n0wall is an embedded firewall distribution of FreeBSD,
- It provides a small image which can be put on Compact Flash cards as well as on CD-ROMs and hard disks.
- m0n0wall provides for a web-based configuration and uses PHP exclusively for the GUI and bootup configuration.

M0n0wall Features

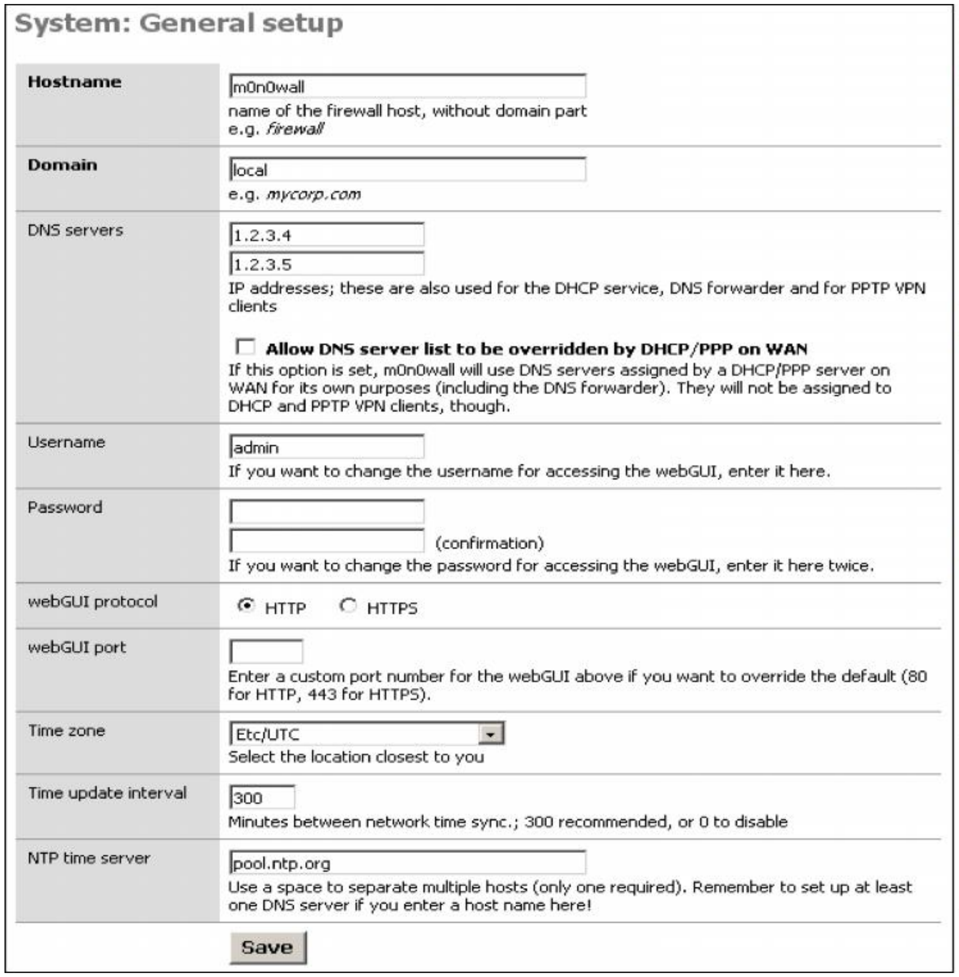
- web interface (supports SSL)
- serial console interface for recovery
- wireless support (including access point mode)
- captive portal
- 802.1Q VLAN support
- IPv6 support
- stateful packet filtering
- NAT/PAT (including 1:1)
- DHCP client, PPPoE and PPTP support on the WAN interface

M0n0wall Features

- IPsec VPN tunnels (IKE; with support for hardware crypto cards, mobile clients and certificates).
- PPTP VPN (with RADIUS server support)
- static routes
- DHCP server and relay
- caching DNS forwarder
- DynDNS client and RFC 2136 DNS updater
- SNMP agent

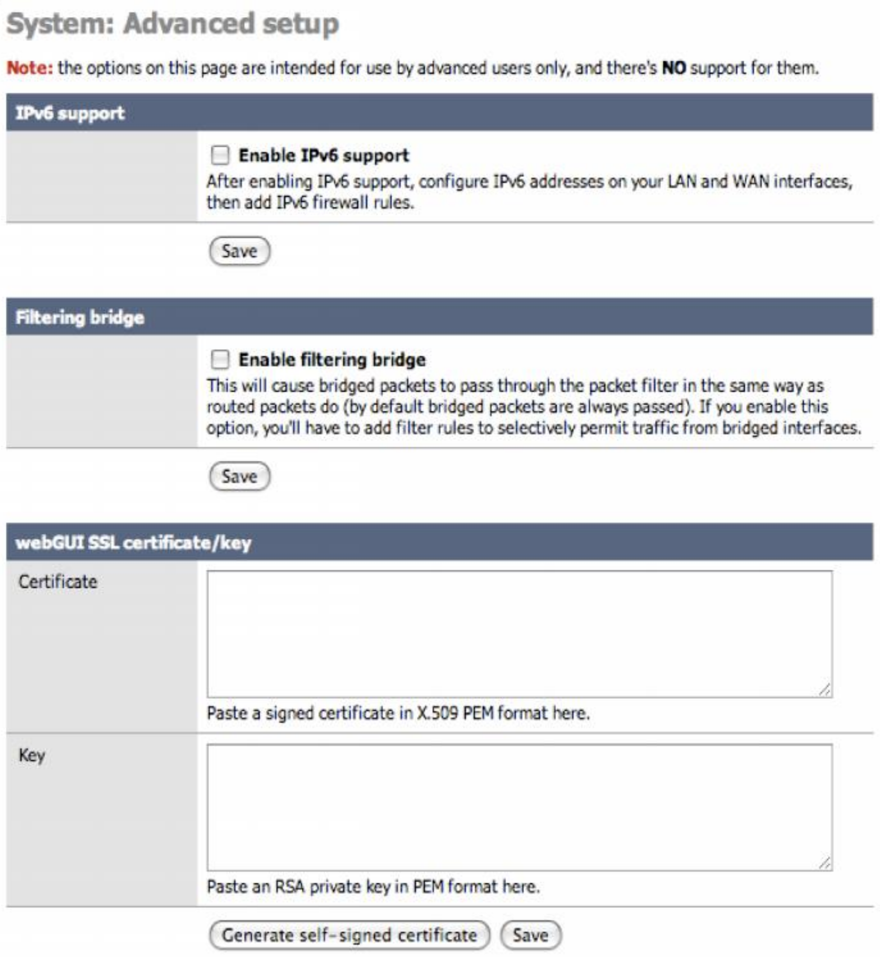
M0n0wall Features

- traffic shaper
- SVG-based traffic grapher
- firmware upgrade through the web browser
- Wake on LAN client
- configuration backup/restore
- host/network aliases



System: General setup

Hostname	m0n0wall name of the firewall host, without domain part e.g. *firewall*
Domain	local e.g. *mycorp.com*
DNS servers	1.2.3.4 1.2.3.5 IP addresses; these are also used for the DHCP service, DNS forwarder and for PPTP VPN clients ☐ **Allow DNS server list to be overridden by DHCP/PPP on WAN** If this option is set, m0n0wall will use DNS servers assigned by a DHCP/PPP server on WAN for its own purposes (including the DNS forwarder). They will not be assigned to DHCP and PPTP VPN clients, though.
Username	admin If you want to change the username for accessing the webGUI, enter it here.
Password	(confirmation) If you want to change the password for accessing the webGUI, enter it here twice.
webGUI protocol	[✓ HTTP] HTTPS
webGUI port	Enter a custom port number for the webGUI above if you want to override the default (80 for HTTP, 443 for HTTPS).
Time zone	Etc/UTC Select the location closest to you
Time update interval	300 Minutes between network time sync.; 300 recommended, or 0 to disable
NTP time server	pool.ntp.org Use a space to separate multiple hosts (only one required). Remember to set up at least one DNS server if you enter a host name here!



System: Advanced setup

Note: the options on this page are intended for use by advanced users only, and there's **NO** support for them.

IPv6 support

☐ **Enable IPv6 support**

After enabling IPv6 support, configure IPv6 addresses on your LAN and WAN interfaces, then add IPv6 firewall rules.

Filtering bridge

☐ **Enable filtering bridge**

This will cause bridged packets to pass through the packet filter in the same way as routed packets do (by default bridged packets are always passed). If you enable this option, you'll have to add filter rules to selectively permit traffic from bridged interfaces.

webGUI SSL certificate/key

Certificate

Paste a signed certificate in X.509 PEM format here.

Key

Paste an RSA private key in PEM format here.