

## Linux.com

Everything Linux and Open Source

### Installing and configuring spamd

April 12, 2007 (8:00:00 AM) - 2 years, 2 months ago

By: **Terrell Prudé, Jr.**

Yesterday **we looked at spamd**, a service designed to reduce the flow of spam to your email inbox. Now that we know some of its advantages, let's put it to work.

The order of things is as follows:

1. Decide on a physical hookup and IP addressing scheme.
2. Get spamd configured and running.
3. Tell OpenBSD to send any TCP 25 traffic to spamd for evaluation (yes, that is a required step). But we also must tell OpenBSD to forward email from "good" senders to our real email server. PF rules (OpenBSD's version of iptables) take care of that.
4. Test, test, test!

I used the physical layout shown in the figure. "spamslayer" is the OpenBSD box running the spamd tarpit/greylister, as well as a stateful packet-filtering firewall. It is a 270MHZ Sun Ultra 5 with 256MB DRAM and a 20GB disk; /var, which holds my logs, is 4GB in size. The box has three interfaces:

1. hme0, the outside interface
2. fxp0, the inside "trusted" interface, and
3. fxp1, the DMZ interface.

"mailhost" is my real mail server. It's another Sun Ultra 5 box running Ubuntu GNU/Linux 6.06 Dapper Drake. It runs Postfix and Dovecot in a maildir configuration, and uses a private IP address of 192.168.10.11.

Once you install and set up the IP addresses on the OpenBSD box, you need to edit three files:

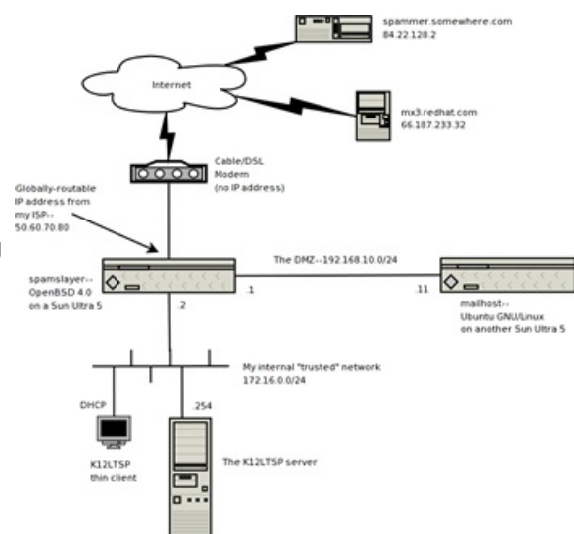
/etc/rc.conf -- the system-wide config file. You can make the changes to /etc/rc.conf.local, if you prefer that instead; the settings in that file override the defaults in /etc/rc.conf.

/etc/pf.conf -- the PF ruleset; mandatory for anything to work

/etc/spamd.conf -- the default is pretty good, but you can make tweaks

A note about interfaces on any of the BSDs: Unlike GNU/Linux, which has eth0, eth1, eth2, and so on, \*BSD interfaces are named by which device driver they take. I have an fxp0 and a fxp1 because I have two Intel EtherExpress Pro/100B cards that use the fxp driver -- equivalent to eepro100 in Linux, and an hme0, which stands for -- no, I'm not kidding -- Sun Happy Meal Ethernet. It's the built-in 10/100 NIC in nearly all Sun Ultra boxes.

/etc/rc.conf controls several boot-time settings. Change the existing lines with these settings so that they now look like the following:



[Click to enlarge](#)

Changes to `/etc/rc.conf`:

```
pf=YES
```

```
spamd_flags="-v -S 90 -n Postfix -h mailhost.cmosnetworks.com -G 60:4:864"
```

```
spamd_grey=YES
```

```
pf_rules=/etc/pf.conf
```

The default time for the initial greylist quarantine is 25 minutes. With a change to the `-G` parameter for `spamd_flags` I bumped it up to 60 minutes, since 25 minutes proved to be a bit too short in actual practice. The `-v` parameter tells `spamd` to log verbosely, so that when we send email to someone, that recipient gets automatically whitelisted.

The `pf_rules` line should already be there, but if it isn't, put it in.

The `spamd` man page will tell you about the rest.

Next up is `/etc/pf.conf`, which I wrote from scratch. If you understand what port forwarding, NAT, and stateful firewalling are, the following file should make sense. Lines that begin with `"#"` are comments:

`/etc/pf.conf`:

```
# First, set up our macros
```

```
externalif = "hme0"
```

```
internalif = "fxp0"
```

```
dmzif = "fxp1"
```

```
internalnet = "172.16.0.0/24"
```

```
dmznet = "192.168.10.0/24"
```

```
# Here are addresses we should never see from the Internet
```

```
# We will use this table to block these IP addresses in a later rule
```

```
table <rfc1918> persist {10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 224.0.0.0/5}
```

```
# Let's make our categories of spammers and non-spammers
```

```
# We do this with PF tables
```

```
table <spamd> persist
```

```
table <spamd-white> persist
```

```
table <whitelist> persist file "/etc/whitelist.txt"
```

```
table <blacklist> persist file "/etc/blacklist.txt"
```

```
# Let's normalize our packets; this is a really good idea
```

```
scrub on $externalif all fragment reassemble random-id reassemble tcp
```

```
# Turn on one-to-many NAT
```

```
nat on $externalif inet proto {tcp, udp, icmp} from $internalnet to any ->
```

```
$externalif
```

```
nat on $externalif inet proto {tcp, udp, icmp} from $dmznet to any -> $externalif
```

```
# Stop the spammers!
```

```
# Redirect SMTP traffic to either our local spamd or the real mail server,
```

```
# depending on which PF table the sender's IP address is in.
```

```
# "Redirect" is OpenBSD PF-speak for Port Address Translation (PAT).
```

```
rdr pass on $externalif proto tcp from <whitelist> to $externalif port smtp ->
```

```
192.168.10.11 port smtp
```

```
rdr pass on $externalif proto tcp from <blacklist> to $externalif port smtp ->
```

```
127.0.0.1 port 8025
```

### Watch out for this potential glitch

You will, as root, need to create a file that is not present in the default install, namely `/var/db/spamd`. Issue the command `touch /var/db/spamd`, which, of course, makes the owner `root:wheel`. However, that file needs to be owned by the `spamd` process's owner, which is `_spamd`. I suspect that the OpenBSD team already knows about this and that it will be fixed in a future version, given that their `spamd` man page mentions the permissions issue. But it's easy enough to correct by running `chown _spamd:_spamd /var/db/spamd`.

```

rdr pass on $externalif proto tcp from <spamd> to $externalif port smtp ->
127.0.0.1 port 8025
rdr pass on $externalif proto tcp from <spamd-white> to $externalif port smtp ->
192.168.10.11 port smtp
rdr pass on $externalif proto tcp from !<spamd-white> to $externalif port smtp ->
127.0.0.1 port 8025

# Need to do some PAT for the real mail server so that IMAP and Secure IMAP work
rdr pass on $externalif proto tcp from any to $externalif port 143 ->
192.168.10.11 port 143
rdr pass on $externalif proto tcp from any to $externalif port 993 ->
192.168.10.11 port 993

#Filter out the spoofer, as defined in the previously-created "<rfc1918>" table
block in quick on $externalif inet from <rfc1918> to any

# Turn on stateful packet filtering
# We let back in any traffic whose session originated from the inside
pass out quick on $externalif inet proto tcp from $internalnet to any modulate
state
pass out quick on $externalif inet proto udp from $internalnet to any keep state
pass out quick on $externalif inet proto icmp from $internalnet to any keep state
# Also have to explicitly allow the firewall's own traffic to come back in!
pass out quick on $externalif inet proto tcp from $externalif to any modulate
state
pass out quick on $externalif inet proto udp from $externalif to any keep state
pass out quick on $externalif inet proto icmp from $externalif to any keep state

# We also let in any SMTP and SSH traffic, and log the SMTP traffic for spamlogd
pass in log quick on $externalif inet proto tcp from any to 192.168.10.11 port
smtp keep state
pass in log quick on $dmzif inet proto tcp from 192.168.10.11 to any port smtp
keep state
pass in quick on $externalif inet proto tcp from any to $externalif port 22

# Deny everything else!
block in on $externalif inet all

End /etc/pf.conf

```

You can tweak the interface names and use this file in your own three-interface OpenBSD box. Note that there are no IP addresses listed here except for that of mailhost (192.168.10.11), which I'm using NAT for anyway. That's because PF allows you to specify interface names instead of IP addresses.

The section labeled "Stop the spammers!" deserves some attention. The order of the statements is important here; if the order is wrong, things won't work right.

First, there's the `<whitelist>` rule. `<whitelist>` is a table of IP addresses and/or netblocks; tables are how you group a bunch of IP addresses together in PF. This particular table is a manual whitelist, not maintained automatically by OpenBSD; it gets populated with the contents of a file I made, `/etc/whitelist.txt`. I keep this list to an absolute minimum. I have four IP addresses in there that I don't want to bother greylisting. Three are on my own network, and the fourth is that of a friend in China. If I ever need to add anything, the system will see it at next reboot. To have OpenBSD recognize an added entry without rebooting, run as root the command `pfctl -t whitelist -T replace -f /etc/whitelist.txt`.

Alternately, you could avoid replacing the entire whitelist table and just add the one that you want to whitelist:

```
pfctl -t whitelist -T add 1.2.3.4
```

or, for a whole 255.255.255.0 network:

```
pfctl -t whitelist -T add 1.2.3.0/24
```

Next is the `<blacklist>` rule. If someone manages to slip by all of spamd's protections, you can put that IP address in here, and it will be tarpitted just as if it were in the SPEWS list. Try to keep the entries in this list to a minimum. I make additions to `/etc/blacklist.txt` and use the same syntax that we used with the whitelist to make them recognized:

```
# pfctl -t blacklist -T replace -f /etc/blacklist.txt
```

Here's what my blacklist file looks like:

```
# Tarpit/reject the mail server of a person who regularly tries to spam me
216.27.93.120
# Tarpit/reject a /8 in China that my other lists missed
121.0.0.0/8
# Tarpit/reject Iquiero.com because they spam me; they use a /24
201.230.255.0/25
```

Whitelists take the same format.

Next comes the `<spamd>` rule. These are the spammers that SPEWS and others know about. Once we're done with everything, they'll all end up in the "spamd" PF table that we created in `/etc/pf.conf` above. This will result in spamd tarpitting them immediately upon connection.

The `<spamd-white>` rule follows that. This basically means, "everyone who got greylisted, but who has passed my initial 60-minute greylisting quarantine, I think you're probably legit, so just head on over to the real mail server." These folks are stored in the "spamd-white" table.

The `!<spamd-white>` section is the catch-all. It means, "everyone else, you're not a known spammer, but you're not whitelisted anywhere either, so you're getting greylisted. Come on back in 60 minutes and I'll let you send email to the real MTA here."

### **Next: /etc/spamd.conf configuration**

Finally, we deal with `/etc/spamd.conf`. If this file is not correct, spamd will complain in various ways -- it may not start, or it might ignore some or all of your configuration. One strong caution: you cannot use tab characters as whitespace in this file; you must use spaces. I tore my hair out trying to figure out why my `/etc/spamd.conf` wasn't working as I expected. I replaced the tab characters with spaces, and everything started working correctly.

Here is my entire `/etc/spamd.conf` file:

```
all:\
    :spews1:spews2:spamhausdroplist:china:korea:russia:mycustom-black:

# Mirrored from http://www.spews.org/spews_list_level1.txt
spews1:\
    :black:\
    :msg="SPAM. Your address %A is in the spews level 1 database\n\
    See http://www.spews.org/ask.cgi?x=%A for more details":\
    :method=http:\
    :file=www.openbsd.org/spamd/spews_list_level1.txt.gz:

# Mirrored from http://www.spews.org/spews_list_level2.txt
spews2:\
    :black:\
    :msg="SPAM. Your address %A is in the spews level 2 database\n\
    See http://www.spews.org/ask.cgi?x=%A for more details":\
```

```

:method=http:\
:file=www.openbsd.org/spamd/spews_list_level2.txt.gz:

spamhausdroplist:\
:black:\
:msg="SPAM. Your address %A is in the Spamhaus drop list\n\
See http://www.spamhaus.org for more details":\
:method=file:\
:file=/etc/spamhaus.droplist.20071227.txt:

# Mirrored from http://www.ocean.com/chinacidr.txt
china:\
:black:\
:msg="SPAM. Your address %A appears to be from China\n\
See http://www.ocean.com/asianspamblocks.html for more details":\
:method=http:\
:file=www.openbsd.org/spamd/chinacidr.txt.gz:

# Mirrored from http://www.ocean.com/koreacidr.txt
korea:\
:black:\
:msg="SPAM. Your address %A appears to be from Korea\n\
See http://www.ocean.com/asianspamblocks.html for more details":\
:method=http:\
:file=www.openbsd.org/spamd/koreacidr.txt.gz:

# Mirrored from
http://www.completewhois.com/statistics/data/ips-bycountry/rirstats/RU-cidr.txt
russia:\
:black:\
:msg="SPAM. Your address %A appears to be a source of spam email\n\
Please contact your ISP regarding this issue":\
:method=http:\
:file=www.completewhois.com/statistics/data/ips-bycountry/rirstats/RU-cidr.txt:

# My custom blacklist
mycustom-black:\
:black:\
:msg="SPAM. Your address %A is in my spammers list. Please stop.":\
:method=file:\
:file=/etc/blacklist.txt:

End /etc/spamd.conf

```

Every address in here gets populated into the `<spamd>` PF table and thus gets automatically and immediately tarpitted the instant that it connects.

And now, the moment of truth: we are ready to turn on spamd. To do so, run `/usr/libexec/spamd-setup now`; whenever you make any changes to `/etc/spamd.conf`, you must run it again for the changes to take effect. `spamd-setup` does two things:

1. It tells PF what to do by loading every IP address or netblock referenced in any of the files mentioned in `/etc/spamd.conf` into the `<spamd>` table defined in `/etc/pf.conf`. Remember that we have a PF rule saying that if an address is in the `<spamd>` table, redirect it to spamd on TCP 8025 on localhost.

2. It tells spamd itself that, in the `/etc/spamd.conf` file, every IP address or netblock in the following lists is to be treated in blacklist mode once PF forwards that address's SMTP traffic to spamd's tarpit:
  - [www.openbsd.org/spamd/spews\\_list\\_level1.txt.gz](http://www.openbsd.org/spamd/spews_list_level1.txt.gz)
  - `/etc/spamhaus.droplist.20071227.txt`
  - [www.openbsd.org/spamd/chinacidr.txt.gz](http://www.openbsd.org/spamd/chinacidr.txt.gz)
  - [www.openbsd.org/spamd/koreacidr.txt.gz](http://www.openbsd.org/spamd/koreacidr.txt.gz)
  - [www.completewhois.com/statistics/data/ips-bycountry/rirstats/RU-cidr.txt](http://www.completewhois.com/statistics/data/ips-bycountry/rirstats/RU-cidr.txt)
  - `/etc/blacklist.txt`

Since there are regular updates to the SPEWS Level 1 and Level 2 lists, you should fire off a cron job once a day that runs `spamd-setup` to download new files and put them into the `<spamd>` table for you.

Note that I added three sections to my `spamd.conf` file in order to tarpit and reject even more spammers than the default configuration does. I patterned these new `spamhausdroplist`, `ruussia`, and `mycustom-black` sections on existing ones in `/etc/spamd.conf`. You can also include the worst offenders from `completewhois.com`, which contains all the netblocks in the world, listed by country.

### **How well does it work?**

Spamd has exceeded my expectations. The spam count in my mailbox has gone down to less than 2% of the amount I was receiving before I began running the utility. I've seen a total of only 10 spam messages over the last 10 days; that's a drop from more than 150 per day. Most days my inbox sees no spam whatsoever. My real mail server (mailhost), which does not run anything like SpamAssassin, is cruising along beautifully, and I continue to get all legitimate email.

Spamd's greytrapping is phenomenal and a significant part of why I get that huge reduction. My spamd logs show large numbers of folks attempting to spam my single fake email address, and all of them get tarpitted and rejected.

Copyright (c) 2007 C. Terrell Prudé, Jr.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Read in the original layout at: <http://www.linux.com/archive/articles/114261>