

## Linux.com

Everything Linux and Open Source

### Reducing spam with OpenBSD and spamd

---

April 11, 2007 (8:00:00 AM) - 2 years, 2 months ago

By: **Terrell Prudé, Jr.**

We all know about the rampant spam email problem. Nearly all of the potential solutions offered for it are based on the idea of the mail server receiving messages, classifying them as either spam or legitimate, and then processing further (deleting or forwarding messages) as appropriate. The problem with this strategy is that you end up using extra resources on the mail server. Here's a way to get the same result while minimizing resource usage by preventing the spam from reaching the mail server.

For this task you can use the **OpenBSD** platform, for two chief reasons:

1. OpenBSD is secure and solid.
2. OpenBSD comes with a tool to stop the spam before it even gets sent: **spamd**, a "fake" Simple Mail Transfer Protocol (SMTP) server that accepts SMTP connections and decides whether a sender is a spammer or not.

Spamd is not an email content analyzer; it does not actually examine a given email's payload. What spamd does is determine -- before the email is ever allowed to be sent in the first place -- whether *the sender itself* is a spammer or not. Spamd sits in front of your real mail server and listens for SMTP conversations. Since it operates at the SMTP level, it will work with any and all **RFC 2821**-compliant SMTP MTAs, including sendmail, Postfix, exim, qmail, and even Microsoft Exchange Server. It also is a good complement to email content analyzers like SpamAssassin.

One key method that spamd uses is a relatively new one called **greylisting**, which initially rejects all email with an SMTP error of 451, which means "temporary failure, please try again later." Properly behaving SMTP gateways will indeed try again later, in accordance with the rules of **RFC 2821**. However, virtually all spammer operations do not try again in order to maximize the amount of spam they spew out.

The greylisting technique returns a 451 error code to any mail sender that spamd doesn't yet know about, and will continue to return that error code to that same sender for a configurable time (25 minutes by default) based on three fields:

1. the sender's IP address,
2. the From: field in the email header, and
3. the To: field in the email header.

This trio, called a tuple, gets put in quarantine, so if the sender does retry before the 25-minute quarantine is over, it will continue to get the 451 error. At the expiration of the quarantine, that sender gets put on a temporary "whitelist" lasting for several hours (four hours by default, including the original 25 minutes). Well-behaved MTAs that try again will be successful, and spamd will pass the email through to your real MTA.

Spamd can also handle known spammers by "tarpitting," or slowing down their connection in order to waste the spammer's time. When a known spammer tries to send you email, spamd will decrease the TCP window length to one to slow the connection down to one byte per second and will not let the connection go. If it gets caught in the tarpit, a spammer's MTA can take as long as 10 minutes to send one message. I regularly see spammers get tarpitted for between 400 and 550 seconds (6 1/2 minutes and just over 9 minutes).

What's worse (for spammers) is that spamd, after wasting all that time, never does allow the spam email through to the real SMTP server. Instead, it sends back a 450 "mailbox busy" message. The spammer retries, and retries, and retries, getting stuck in the tarpit every time. I had one spammer that kept retrying -- and repeatedly getting stuck in my trap -- for a day and a half, and never once was that spammer able to actually transmit the spam message to me.

Tarpitting can be implemented for senders on the same **SPEWS/Spamhaus** blacklist that you're likely using with a different antispam tool. Spamd's default configuration automatically tarpits the following IP addresses:

- any IP netblocks in either the SPEWS Level 1 or SPEWS Level 2 lists
- any IP netblocks in China
- any IP netblocks in Korea

The reason for the SPEWS lists is obvious. China and Korea are blocked because so many spam email servers are located in those two countries, and their ISPs and governments don't seem to have any interest in getting rid of them. I also have added all of Russia's IP netblocks to my configuration, for the same reasons.

If you need to direct non-spammers who are in a blacklisted network past spamd, you can add their mail servers to a permanent whitelist that gets processed before any greylisting or blacklisting occurs. This allows their mail servers to bypass the greylisting and blacklisting functions and go straight to your real mail server.

There's one other handy thing that spamd can do for us. Spamd can optionally monitor your mail logs and automatically whitelist the destination email servers of anyone to whom you send email.

Another additional optional feature of greylisting with OpenBSD is something called greytrapping. Spammers "harvest" anything that looks like an email address from Web pages throughout the Internet, looking for potential victims. If you post a fake email address on your site that does not actually exist on your real email server, you'll know that if someone tries to send email

to that fake email address, it's a spammer. Spamd checks the recipient in the SMTP "RCPT TO:" information against a list of fake recipient email addresses that you've previously told it to watch for. If it sees fakeaddress@mydomain.com, it immediately tarpits the mail server's IP address.

There's not much not to like in spamd. How do you get it to work? We'll tackle that tomorrow.

Read in the original layout at: **<http://www.linux.com/archive/articles/61103>**