Using Greylist with FreeBSD

Tom Rhodes <trhodes@FreeBSD.org>
Revision: 46449

2015-04-03 15:30:41Z by eadler.

Abstract

An article written for the sole purpose of explaining the relaydelay system on a FreeBSD mail server. A relaydelay or greylisting server cuts down on spam simply by issuing a TEMPFAIL error message to every incoming email. The purpose behind this idea is that most spammers use their personal computers with software to do their spamming. A real mail server should queue the message and try to send it later. Thus the spammer most likely moves on to the next host in place of trying to send the email again. This is an excellent idea; at least until the spammers begin to use software that offers to try again. But how does this work exactly? Well, when an email is received the message ID is stored in a database and the TEMPFAIL is returned along with the email. If the email is resent, the message ID will be checked against the message IDs currently stored in the database. If it exists in the database then the email is permitted to reach its intended recipient. Otherwise, the ID will be stored and a TEMPFAIL will be issued. This cycle will repeat with every email which comes into the server. From my personal experience, this really does cut out 90% of the spam.

Table of Contents

1. Basic Configuration

Install perl using

```
# pkg install lang/perl5.16
```

Now for the database server; MySQL is perfect for this sort of work. Install the databases/mysql40-server along with databases/p5-DBD-mysql40. The previous port should imply the installation of databases/p5-DBI-137 so that knocks off another step.

Install the `perl` based portable server plugin, net/p5-Net-Daemon port. Most of these port installations should have been straight forward. The next step will be more involved.

Now install the mail/p5-Sendmail-Milter port. As of this writing the `Makefile` contains a line beginning with `BROKEN`, just remove it or comment it out. It is only marked this way because FreeBSD neither has nor installs a threaded `perl` package by default. Once that line is removed it should build and install perfectly fine.

Create a directory to hold temporary configuration files:

```
# mkdir /tmp/relaydelay
# cd /tmp/relaydelay
```

Now that we have a temporary directory to work in, the following URLs should be sent to the `fetch` command:

```
# fetch http://projects.puremagic.com/greylisting/releases/relaydelay-0.04.tgz
# fetch http://lists.puremagic.com/pipermail/greylist-users/attachments/20030904/b8dafed9/
relaydelay-0.04.bin
```

The source code should now be unpacked:

```
# gunzip -c relaydelay-0.04.tgz | tar xvf -
```

There should now be several files into the temporary directory by this point. The appropriate information can now be passed to the database server by importing it from the `mysql.sql` file:

```
# mysql < relaydelay-0.04/mysql.sql
```

And patch the other files with the `relaydelay.bin` by running:

```
# patch -d /tmp/relaydelay/relaydelay-0.04 < relaydelay.bin
```

Edit the `relaydelay.conf` and the `db_maintenance.pl` file to append the correct username and password for the MySQL database. If the database was built and installed like the above then no users or passwords exist. This should be altered before putting this into production, that is covered in the database documentation and is beyond the scope of this document.

Change the working directory to the `relaydelay-0.04` directory:

```
# cd relaydelay-0.04
```

Copy or move the configuration files to their respective directories:

```
# mv db_maintenance.pl relaydelay.pl /usr/local/sbin
# mv relaydelay.conf /etc/mail
# mv relaydelay.sh /usr/local/etc/rc.d/
```

Test the current configuration by running:

```
# sh /usr/local/etc/rc.d/relaydelay.sh start
```

Note

This file will not exist if the previous mv(1) commands were neglected.

If everything worked correctly a new file, `relaydelay.log`, should exist in `/var/log` . It should contain something similar to the following text:

```
Loaded Config File: /etc/mail/relaydelay.conf
Using connection 'local:/var/run/relaydelay.sock' for filter relaydelay
DBI Connecting to DBI:mysql:database=relaydelay:host=localhost:port=3306
Spawned relaydelay daemon process 38277.
Starting Sendmail::Milter 0.18 engine.
```

If this does not appear then something went wrong, review the screen output or look for anything new in the `messages` log file.

Glue everything together by adding the following line to `/etc/mail/sendmail.mc` or the customized site specific `mc` file:

```
INPUT_MAIL_FILTER(`relaydelay', `S=local:/var/run/relaydelay.sock, T=S:1m;R:2m;E:3m')dnl
```

Rebuild and reinstall the files in the `/etc/mail` directory and restart `sendmail`. A quick `make restart` should do the trick.

Obtain the `perl` script located at http://lists.puremagic.com/pipermail/greylist-users/2003-November/000327.html and save it in the `relaydelay-0.04` directory. In the following examples this script is referred to as `addlist.pl`.

Edit the `whitelist_ip.txt` file and modify it to include IP addresses of servers which should have the explicit abilities to bypass the relaydelay filters. i.e., domains from which email will not be issued a TEMPFAIL when received.

Some examples could include:

```
192.168.   # My internal network.
66.218.66       # Yahoo groups has unique senders.
```

The `blacklist_ip.txt` file should be treated similarly but with reversed rules. List within this file IPs which should be denied without being issued a TEMPFAIL. This list of domains will never have the opportunity to prove that they are legitimate email servers.

These files should now be imported into the database with the `addlist.pl` script obtained a few lines ago:

```
# perl addlist.pl -whitelist 9999-12-31 23:59:59 < whitelist_ip.txt
# perl addlist.pl -blacklist 9999-12-31 23:59:59 < blacklist_ip.txt
```

To have relaydelay start with every system boot, add the `relaydelay_enable="YES"` to the `/etc/rc.conf` file.

The `/var/log/relaydelay.log` log file should slowly fill up with success stories. Lines like the following should appear after a short time, depending on how busy the mail server is.

```
=== 2004-05-24 21:03:22 ===
Stored Sender: <someasshole@flawed-example.com>
Passed Recipient: <local_user@pittgoth.com>
  Relay: example.net [XXX.XX.XXX.XX] - If_Addr: MY_IP_ADDRESS
  RelayIP: XX.XX.XX.XX - RelayName: example.net - RelayIdent:  - PossiblyForged: 0
  From: someasshole@flawed-example.com - To: local_user
  InMailer: esmtp - OutMailer: local - QueueID: i4P13Lo6000701111
  Email is known but block has not expired.  Issuing a tempfail.  rowid: 51
  IN ABORT CALLBACK - PrivData: 0<someasshole@flawed-example.com>
```

The following line may now be added to `/etc/newsyslog.conf` to cause for `relaydelay.log` rotation at every 100 Kb:

```
/var/log/relaydelay.log                 644  3     100  *     Z
```

Note

At some point there was an error about improper `perl` variables in the `/etc/mail/relaydelay.conf`. If those two variables are commented out then configuration may proceed as normal. Just remember to uncomment them before starting the `relaydelay` process.