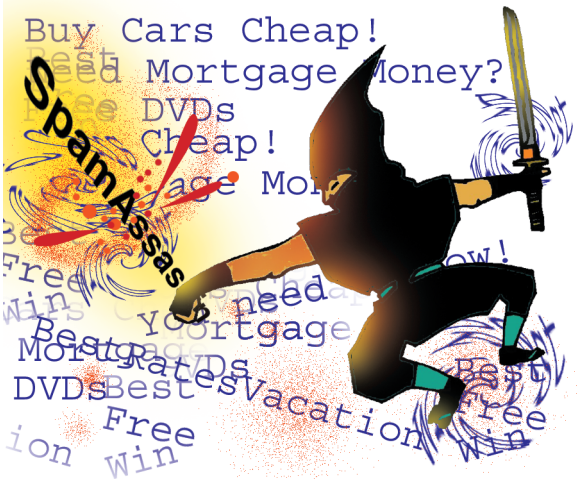
Integrating SpamAssassin into Qmail and Sendmail

Jason Camp

Spam is an increasing problem for many people, including systems administrators. According to a study by Ferris Research, in 2002 U.S. and European businesses spent more than $11 billion combating the overwhelming amount of junk email. Brightmail Inc. estimated that in 2002 approximately 40% of all Internet mail was some form of unsolicited bulk email. While there are some laws in place to prevent fraudulent junk email in the U.S., more than 50% of spam is sent from non-U.S. and non-European countries. Companies have turned to specialized solutions for Qmail and Sendmail to reduce wasted resources and the amount of unsolicited email. This article will discuss integration with SpamAssassin, an open source solution to this problem.

SpamAssassin is a filtering tool that employs a wide range of heuristic tests to determine whether a piece of mail should be classified as junk email. Common identifiers of spam, such as invalid headers and suspicious phrases, will result in higher scoring. The score is lower if SpamAssassin encounters legitimate email headers. There is also support for Bayesian filtering, black hole lists, such as the Realtime Blackhole List (mail-abuse.org), SpamCop (bl.spamcop.net), and Razor (razor.sourceforge.net). Utilizing any number of these features can present a formidable layer of spam protection.

When implementing any type of email filtering, you must determine at what point the mail is to be filtered. Preprocessing the email when it is received has the benefit of utilizing the server's resources instead of client resources. It allows for a single software installation and central management, as opposed to maintaining multiple procmail scripts for each user, or individual client software for POP or IMAP users. When preprocessing mail, keep in mind that if you are providing email services to multiple users, there will always be some users who do not want to have their mail filtered. My solutions for Qmail and sendmail provide the ability to select the users that are to be filtered.



Installing SpamAssassin

SpamAssassin consists of three main components: a set of Perl modules, a Perl daemon (spamd), along with a C client (spamc). Perl 5.6 or higher is recommended. There are two CPAN modules that are required that are not included in the standard Perl distribution — HTML::Parser and Sys::Syslog.

The latest version of SpamAssassin (as of this writing) is 2.53. Download the tarball from the SpamAssassin site:

```
http://www.spamassassin.org/released/Mail-SpamAssassin-2.53.tar.gz
```

Install the HTML::Parser module through CPAN or manual install:

```
#  perl -MCPAN -e 'install HTML::Parser'
#  perl -MCPAN -e 'install Sys::Syslog'

Install SpamAssassin:
# tar xzf Mail-SpamAssassin-2.53.tar.gz
# cd Mail-SpamAssassin-2.53
# ./configure
# make
# make install
```

What to Do with Tagged spam

Before configuring your mail server, you must decide what will happen to the mail that SpamAssassin tags. You could simply delete it, tag it and pass it along, or inject it into another folder (like a "Bulk Mail" folder).

While SpamAssassin is 99% effective in finding and tagging spam, it will produce false positives over time. For example, when I applied for a mortgage online, SpamAssassin tagged a few of the response emails because of the mortgage heuristic filters. I recommend that you never delete tagged mail unless you have a good reason for it, and can live with the possibility of valid emails occasionally getting deleted. The simplest approach is to allow your MTA to continue to deliver the tagged mail. Mail will still be placed in the user's inbox, but it is much easier to deal with once the "***** SPAM *****" header is added to the subject line.

HTML-formatted emails will also be converted into plaintext, avoiding potential HTML viruses and Web bugs. Filtering tagged mail into another mailbox is another alternative. This is the ideal solution, but does require some per-user configuration, and could result in a damaged mailbox depending on the mailbox format you are using and the method of injection. If you have multiple email accounts and are filtering for yourself, you may want to have all of the tagged mail dumped into one "catch-all" email address that you could sift through.

Configuring SpamAssassin for Qmail

SpamAssassin keeps its site configuration in the /etc/mail/spamassassin/local.cf file. By default, SpamAssassin employs a set of local heuristic tests, which are sufficient for catching most spam. You can tailor these to your liking once everything is up and running.

When filtering mail, you can either call SpamAssassin directly, or set up the spamd daemon and use the spamc client. The client/daemon is much faster than calling SpamAssassin directly, since Perl needs to load up a copy for every piece of mail that comes in. The spamc client is written in C, which makes the overhead per email very light. If you are configuring a qmail mail server, set up the spamd daemon. The sendmail milter integrates directly into the SpamAssassin Perl libraries and no daemon is required.

Spamd takes a few different arguments. You must specify the user that runs spamd, the servers that are allowed to connect, and to ignore individual user config files. Place this line in your qmail init script as the first line in the start subroutine:

```
# /usr/local/bin/spamd 0 -L -x -u nobody -A 10.1.1.161 &
```

Qmail Configuration

Qmail is an extensible mail server, and through the use of dot qmail files you have the ability to forward tagged messages, drop them into another mailbox, or send to a catchall address. For per-user configuration, dot qmail files and a simple shell script are all you need. James Grinter has written a shell script called ifspamh that will process the mail through spamc. It will re-inject the mail to another email address if it is tagged, or continue through the dot qmail file if it is not tagged. The ifspamh script can be downloaded at http://www.gbnet.net/~jrg/qmail/ifspamh/ifspamh. You need to modify the locations of the SPAMC and INJECT variables if you have installed SpamAssassin or qmail in alternate locations. Ifspamh also requires the mess822 utilities from DJB at: http://cr.yp.to/mess822.html.

If you want to check incoming mail, tag the spam, and drop it into a BulkMail folder, you need to set up two dot qmail forward files. The first file sends the email through the parser and handles the mail if it is not tagged, and the second file handles the tagged mail. This example will forward the mail to the account jcamp-sa@example.com if it is tagged as spam, or drop it into the root Maildir of the jcamp user if it is not. When it is re-injected to jcamp-sa@example.com, it will read the .qmail-jcamp-sa file and drop it into the .BulkMail folder:

```
# cat .qmail-jcamp
|/usr/local/bin/ifspamh jcamp-sa@example.com
/home/jcamp/Maildir/

# cat .qmail-jcamp-sa
/home/jcamp/Maildir/.BulkMail/
```

If you would like to tag incoming junk email, but still have all mail go into the inbox, you still need to set up two dot qmail files. It is a little convoluted, but it allows qmail to do the work instead of relying on external tools to re-inject the tagged mail. This example is similar to the first, but in this scenario both dot qmail files drop the mail into their root Maildir folder:

```
# cat .qmail-jcamp
|/usr/local/bin/ifspamh jcamp-sa@example.com
/home/jcamp/Maildir/

# cat .qmail-jcamp-sa
/home/jcamp/Maildir/
```

If you would like to delete the mail if it is marked as spam, create both dot qmail files and put a single # in the second file. Qmail will read this as a comment, and just drop the mail into the abyss:

```
# cat .qmail-jcamp
|/usr/local/bin/ifspamh jcamp-sa@example.com
/home/jcamp/Maildir/

# cat .qmail-jcamp-sa
#
```

Sendmail Configuration

Sendmail 8.12.0 (and later) includes support for milters, which are external programs that can preprocess incoming email. Milter support started in sendmail 8.11, but was for the most part broken until the first release of 8.12. There are a few different milters that support SpamAssassin, but MIMEDefang (http://www.roaringpenguin.com/mimedefang/) is by far the most extensible and supported milter available. MIMEDefang can do much more than just SpamAssassin processing, including virus scanning, deletion of attachments based on file name or content, and rejection of emails that violate your email policies. It allows full integration with the SpamAssassin Perl libraries, which does not require the spamd daemon to be running.

MIMEDefang recommends that you use at least sendmail 8.12.5. Note that if you are running the version of sendmail that came with your OS, you will probably need to recompile sendmail to incorporate milter support. See the sidebar for basic instructions on compiling sendmail from a source tar, and recompiling the sendmail source RPM under Red Hat 8.0.

The steps to get Red Hat's installation of sendmail working with milter are tedious and the sidebar should save you some trouble.

There are many Perl modules required, and these are listed below (HTML::Tagset and HTML::Parser are required, but should have been installed with SpamAssassin). The latest version of MIMEDefang as of this article is 2.30, located at:

```
http://www.roaringpenguin.com/mimedefang/mimedefang-2.30.tar.gz
```

Milter Configuration

Add the milter line into the /etc/mail/sendmail.mc file:

```
INPUT_MAIL_FILTER('mimedefang','S=unix:/var/spool/MIMEDefang/ \
  mimedefang.sock, T=S:5m;R:5m')
```

Rebuild the sendmail.mc file:

```
# cd /etc/mail
# make
# m4 sendmail.mc > /etc/mail/sendmail.cf
```

Install the CPAN perl modules through CPAN or manual install:

```
# perl -MCPAN -e 'install IO::Stringy'
# perl -MCPAN -e 'install MIME::Tools'
```

```
# perl -MCPAN -e 'install M/MA/MARKOV/MailTools-1.58.tar.gz'
# perl -MCPAN -e 'install MIME::Base64'
# perl -MCPAN -e 'install Digest::SHA1'
# perl -MCPAN -e 'install Bundle::libnet'
# perl -MCPAN -e 'install Mail::Audit'
# perl -MCPAN -e 'install Time::HiRes'
```

Installing MIMEDefang

First, add the defang user:

```
# useradd defang
# mkdir /var/spool/MIMEDefang
# chmod 700 /var/spool/MIMEDefang
# chown defang:defang /var/spool/MIMEDefang
```

Configure and install MIMEDefang:

```
# tar xzf mimedefang-2.30.tar.gz
# cd mimedefang-2.30
# ./configure
# make
# make install
# cp examples/init-script /etc/init.d/mimedefang
# touch /etc/mail/spamcheck.users
```

Add the following lines to the /etc/init.d/sendmail script:

```
/etc/init.d/mimedefang start
```

at the beginning of the start() subroutine, and:

```
/etc/init.d/mimedefang stop
```

at the end of the stop() subroutine.

Modify the queue runner so it checks the queue every five minutes. If there are multiple local recipients listed in an incoming email, the mimedefang-filter will split the email into two separate mails — one for those users that require filtering, and one for the users that do not. Sendmail accepts these new emails as deferred; if you change the default from 1 hour to 5 minutes, this will cause these mails to be processed shortly after they are resent. Edit /etc/sysconfig/sendmail and set the QUEUE variable to 5 minutes:

```
QUEUE=5m
```

The default MIMEDefang filter file, located in /etc/mail/mimedefang-filter, requires some hefty changes. You can view and download my custom filter file at `http://www.samag.com/code/`. I have pulled some of the default configuration out that deals with virus checking, and added a few subroutines to allow for per-user checking. There are also code snippets for changing the subject line (MIMEDefang communicates directly with SpamAssassin, and prevents SpamAssassin from actually making changes to the email), adding the report into the body of the email, and converting all text/html MIME types into text/plain.

Only users that reside in the /etc/mail/spamcheck.users file will actually be processed by SpamAssassin, all other users will not be filtered. This could be extended into other parts of MIMEDefang as well, to allow attachment virus scanning or other MIMEDefang features.

Conclusion

I have been running similar configurations for a few mail servers without any performance problems. Through the use of the spamd daemon with qmail, and the MIMEDefang milter with sendmail, there is very little overhead per message. Getting the sendmail install to properly recognize the milter libraries was not fun, but once it was working things went smoothly. This provides a simple approach to dealing with the overwhelming spam problem that seems to continue growing. The time spent integrating SpamAssassin into my mail server was well worth it considering all the time saved weeding out the spam to get to my real email.

Jason Camp has worked as a systems administrator for more than eight years, working for Fortune 500 companies, colleges, and dot coms. He has owned and operated an ISP and Web hosting company for six years. He is Sun Solaris 8 Certified, and works with a number of platforms, including Linux, FreeBSD, and Solaris. He can be contacted at: jcamp@vhosting.com.

Sendmail Source Installation

Download the latest sendmail source tar at:

```
ftp://ftp.sendmail.org/pub/sendmail/
```

The latest release, as of this article, is 8.12.9. Untar:

```
# tar xzf sendmail-8.12.9.tar.gz
```

Sendmail does not support milters by default, so you must create the file devtools/Site/site.config.m4 and add the following lines:

```
dnl Milter
APPENDDEF('conf_sendmail_ENVDEF', '-DMILTER')
APPENDDEF('conf_libmilter_ENVDEF', '-D_FFR_MILTER_ROOT_UNSAFE')
```

Cd to the sendmail directory and run the build script:

```
# cd sendmail
# sh Build
# sh Build install
```

Copy in a few things from the sendmail compile. Your obj* directory may differ depending on your kernel version and architecture, so substitute the appropriate values:

```
# mkdir -p /usr/local/include/sendmail
# cp -R include/* /usr/local/include/sendmail
# cp -R sendmail/*.h /usr/local/include/sendmail
# cp obj.Linux.2.4.7-10.i686/*/*.a /usr/local/lib
```

Red Hat 8.0 Installation

The sendmail source rpm for Red Hat 8.0 for the Intel architecture is located at:

```
ftp://ftp.redhat.com/pub/redhat/linux/8.0/en/os/i386/SRPMS/ \
  sendmail-8.12.5-7.src.rpm
```

You may have to satisfy a number of development dependencies, which can be removed once the binary rpm is built. In my installation, for example, openssl-devel, db4-devel, and sendmail-cf were required but not listed in the rpm dependency list:

```
# rpm -ivh sendmail-8.12.5-7.src.rpm
# cd /usr/src/redhat/SPECS
# rpmbuild -bc sendmail.spec
```

Copy in a few things from the sendmail compile. Your obj directory may differ depending on your kernel version and architecture, so substitute the appropriate values:

```
# cp -r /usr/src/redhat/BUILDS/sendmail-8.12.5/include/ \
  libmilter /usr/local/include
# cp /usr/src/redhat/BUILDS/sendmail-8.12.5/ \
  obj.Linux.2.4.18-14.i686/libmilter/libmilter.a /usr/local/lib
```