Table of Contents

Scope

The scope of this report is to discuss setting up a mail server with smart spam filtering. This report will not explain with any depth how to setup a mail server when it comes to basic setup such as routing mail and similar basic message delivery agent(MDA) tasks but it will cover integrating the mail server with the anti spam software.

About the software

This project describes how to install a mail server with spam filtering. The software chosen is:

- Qmail
- Spam Assassin
- Pyzor/Razor2

About Qmail

Qmail is a free mail server written from scratch by Dan Bernstein, a professor at the University of Illinois. It's small, very fast and full-featured. It was written to be completely secure, which is something most software designers will not say, but there is still an unclaimed cash price for discovering a security hole in the software. (Note: This must be a security hole in the Qmail package, it does not apply to poorly configured servers). Have a look at Appendix I to see how each part of the Qmail package fits together.

About SpamAssassin

Spamassassin is a mail filter to identify spam using text analysis and various Internet-based real time blacklists. If a mail is believed to be spam, it will be “tagged” as spam so the mail can be dealt with down the line. Possible actions can be that the tagged message is discarded by the system or the users mail agent filters the message into a spam folder in the users mail agent.

Pyzor/Razor2

Pyzor and Razor2 are frameworks that allow individuals to query large spam databases. The clients also allow reporting spam to update the database. This is a very good defense against spam because for spammers to gain anything they have to send spam to very large numbers of individual users. Pyzor started as a Python (programming language) implementation of Razor, but due to the protocol and the fact that Razor's server is not Open Source or software libre Pyzor was released with a new protocol and is Open Source GPL

License Issues

Qmail

Qmail is copyrighted by the author, Dan Bernstein, and is released under the OpenContent License (OCL). This license restricts redistribution of the Qmail package so that no changes can be made to Qmail and distributed as a package. This means that Qmail's users that need special features or bug fixes have to patch the Qmail package them selves. The bottom line is that only Dan Bernstein can make changes to the Qmail package.

The full license is at http://opencontent.org/opl.shtml

SpamAssassin

SpamAssassin is written in Perl and is released under the same license as Perl itself. Perl is distributed under under the GNU Public License, published by the Free Software Foundation, or a “Artistic License” which is distributed with Perl. The user can choose which license he wants to adhere to. SpamAssassin is trademark of Deersoft Inc.

Pyzor/Razor2

Pyzor is fully Open Source GPL. Both clients and server software, this means anyone can setup it's own Pyzor servers. Razor2's clients are released under Perl's “artistic” license, which means anyone can use the Razor database but no one can set up their own servers.

Setting up Qmail

Some Linux distributions solve the licensing problem by automating the building and patching process so it is almost transparent to users.

Installation

Qmail in Debian Linux

Debian for example downloads the Qmail package and installs a script that will compile the package for you.

```
marvin:~# apt-get install qmail-src
[ fetch output ]
marvin:~# build-qmail
This script unpacks the qmail source into a directory, and compiles it to produce a
binary qmail*.deb file.
The directory where this is done will end up containing the source and package files
for the qmail binary package, along with a directory containing the unpacked source.
!* WARNING *!
There have been reports of undesired behavior when attempting to build qmail in a
directory on a tmpfs based filesystem.  Please do not try to build on an tmpfs
filesystem.
Enter a directory where you would like to do this [/tmp/qmail] /tmp/qmail-test

Binary package qmail will be compiled now
If you want to apply a custom patch, switch to another console and do it now
The following patches have been applied for you automagically:
netscape-progress - Fixes compatibility bug in POP3 daemon
pop3-supplementarygroups - Update to checkpasswd to allow multiple groups
qmail-link-sync - Filesystem performance patch
qmail-smtpd-bmtpatch - Implements badmailto functionality
qmailqueue - Allows use of external qmail-queue programs
qregex - Allows use of regular expressions for anti-relay / spam control
errno - Fixes glibc compatibility error
qmail_local - Fixes memory corruption in certain .qmail files
Patches already applied can be found in the patches/ directory where the qmail-src
package was extracted.
This can take long time, depending on your machine
Press ENTER to continue...
```

Note: Debian has a precompiled package called `qmail`, but we will cover `qmail-src` here.

This script unpacks the source code to a directory of your choosing and applies various patches on the source code tree. These patches are widely accepted by the community, but not accepted by Dan Bernstein and therefor have not been fixed in the Qmail package.

If the user wishes to apply any more patches, or remove (to reverse) some of the applied patches he can switch to a different terminal and do this. When he is satisfied with the source tree he returns to the terminal and presses enter so the script begins to compile the Qmail binaries.

```
Press ENTER to continue...
[lot's of compiler output]
dpkg --build debian/tmp ..
dpkg-deb: building package `qmail' in `../qmail_1.03-31_i386.deb'.
It seems that all went ok
Do you want to remove all files in /tmp/qmail-test,
except qmail_1.03-31_i386.deb now? [Yn]
Removing files... done
Do you want to install qmail_1.03-31_i386.deb now? [Yn]
Do you want to purge qmail-src now? [yN]
```

```
Remember that you can install qmail_1.03-31_i386.deb on other computers so you don't
need to compile it again.

Don't forget to setup a /etc/qmail/rcpthosts file to prevent open relaying!

Good luck!
```

And that's all there is to the binary installation on debian.

Qmail in Gentoo Linux

Gentoo Linux also patches the Qmail package automatically for the user as this part of the emerge log shows:

```
>>> Unpacking qmail-1.03.tar.gz to /var/tmp/portage/qmail-1.03-r13/work

 * Adding SMTP AUTH (2 way), Qregex and STARTTLS support
 * Applying smtp-auth-close3.patch...
 * Adding QMAILQUEUE support
 * Patching for large queues
 * Adding support for remote QMTP hosts
 * Adding support for oversize DNS
 * Applying qmail-local-tabs.patch...
 * Applying qmail-link-sync.patch...
 * Applying big-concurrency.patch...
 * Applying qmail-0.0.0.0.patch...
 * Applying errno.patch...
 * Applying sendmail-flagf.patch...
 * Applying qmail-maildir++.patch...
 * Applying qmail-date-localtime.patch.txt...
 * Applying qmail-limit-bounce-size.patch.txt...
 * Applying qmail-smtpd-esmtp-size-gentoo.patch...
 * Applying qmail-smtpd-relay-reject.gentoo.patch...
 * Applying qmail-gentoo-1.03-r12-badrcptto-morebadrcptto-accdias.diff.bz2...
 * Enable stderr logging from checkpassword programs
 * Allow qmail to re-read concurrency limits on HUP
 * Add support for CAPA in POP3d
 * Enabling SSL/TLS functionality
 * Replacing obsolete head/tail with POSIX compliant ones

>>> Source unpacked.
```

Though Gentoo's compilation script does not stop for the user to configure the source tree before compiling the binaries. If a user wants to apply custom patches he can do the following:

```
berbara root # ebuild /usr/portage/net-mail/qmail/qmail-1.03-r13.ebuild unpack
```

This will fetch the needed packages, unpack and patch the source tree. The state of the source tree is no ready to start compiling, so the user can apply or remove patches before continuing by invoking:

```
berbara root # ebuild /usr/portage/net-mail/qmail/qmail-1.03-r13.ebuild merge
```

Which will go through ebuild's compile, install and qmerge stages.

The important output from the qmerge stage is shown here:

```
 * To setup qmail to run out-of-the-box on your system, run:
 * ebuild /var/db/pkg/net-mail/qmail-1.03-r13/qmail-1.03-r13.ebuild config

 * To start qmail at boot you have to add svscan to your startup
 * and create the following links:
 * ln -s /var/qmail/supervise/qmail-send /service/qmail-send
 * ln -s /var/qmail/supervise/qmail-smtpd /service/qmail-smtpd
```

```
 * To start the pop3 server as well, create the following link:
 * ln -s /var/qmail/supervise/qmail-pop3d /service/qmail-pop3d

 * Additionally, the QMTP and QMQP protocols are supported,
 * and can be started as:
 * ln -s /var/qmail/supervise/qmail-qmtpd /service/qmail-qmtpd
 * ln -s /var/qmail/supervise/qmail-qmqpd /service/qmail-qmqpd
```

I will not go into depth about how to start Qmail as distributions vary much in the way Qmail is started, but the normal way to start Qmail is by using the tcpserver wrapper package. Qmail can also run from inetd.

Configuring Qmail

This report's scope does not cover setting up Qmail but I will mention some important files that Qmail needs. All of Qmail's configuration files are in `/var/qmail/control/` except the .qmail files in `~alias`. The following files is provided as an example, the setup is from the wigen.net mail domain. Wigen.net is configured as a virtual domain.

File: /var/qmail/control/me

```
wigen.net
```

The `me` file should contain the fully qualified domain name for the mail server (FQDN). Many configuration files default to the FQDN defined in this file.

File: /var/qmail/control/defaultdomain

```
localhost
```

The `defaultdomain` file defaults to what is defined in `me`. qmail-inject adds this name to any host name without dots, including defaulthost if defaulthost does not have dots.

File: /var/qmail/control/locals

```
localhost
```

The `locals` file defaults to `me` and defines which domains we deliver localy.

File: /var/qmail/control/rcpthosts

```
localhost
mail.wigen.net
www.wigen.net
wigen.net
```

If no `rcpthosts` file exist Qmail will accept mail to any domain. This file is important do define so the mail server will not act as an open relay.

File: /var/qmail/control/smtproutes

```
:mail.online.no
```

The `smtproutes` file has no default value. It defines artificial SMTP routes. Each route has the form `domain:relay`, without any extra spaces. If `domain` matches `host`, qmail-remote will connect to `relay`, as if `host` had `relay` as its only MX. In wigen.net's case all outgoing mail will be routed to wigen.net's ISP's mail server.

```
File: /var/qmail/control/virtualdomains
www.wigen.net:www.wigen.net
wigen.net:wigen.net
```

This is a very brief look at some configuration files for Qmail but it is all I will cover about setting up the default MDA behaviour in this report.

Setting up SpamAssassin

Installation

Installation of SpamAssassin is trivial on most Linux distributions. On Debian a simple

```
marvin:~# apt-get install spamassassin
```

will do the trick. On Gentoo doing:

```
marvin:~# emerge spamassassin
```

will install the needed files. SpamAssassin is also available in CPAN. The command:

```
perl -MCPAN -e shell
```

Will fetch and install the spamassassin module for Perl. Not this will not install the spamc client and spamd deamon so using your Linux distribution's native package installer is recommended.

The SpamAssassin package provides a daemonized version of spamassassin (spamd) which goal is improving throughput performance for automated mail checking. spamd is intended to be used alongside "spamc", a fast, low-overhead client program written in C.

Configuring SpamAssassin

SpamAssassin's default behavior is to tag the message subject and also add some X-SPAM headers to the message if it is thought to be spam.

This listing shows the default global configuration file on Debian:

```
marvin:/etc/spamassassin# cat local.cf
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
###########################################################################
#
# rewrite_subject 0
# report_safe 1
# trusted_networks 212.17.35.
required_hits 6.0
rewrite_subject 1
report_header 1
use_terse_report 1
defang_mime 1
dns_available yes
dcc_add_header 1
use_pyzor 1
use_razor2
use_dcc 1
```

Lines starting with hash(#) are comments.

On Gentoo Linux the SpamAssassin's configuration file is located under `/etc/mail/spamassassin/` but the default configuration file does not change any of SpamAssassin's default behavior like Debian does.

```
berbara spamassassin # cat local.cf
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
###########################################################################
#
# rewrite_subject 0
# report_safe 1
# trusted_networks 212.17.35.
```

SpamAssassin is very well documented, the configuration and the defaults are all documented in the manual page:

```
alex@berbara:~$ man Mail::SpamAssassin::Conf
```

Some interesting configuration options are:

```
ok_languages en no da sv
```

This specifies which languages are considered OK for incoming mail. In this case English, Norwegian, Danish and Swedish are accepted. On large servers with users from many nationalities this is might not be a good approach.

> Note that the language cannot always be recognized with sufficient confidence. In that case, no points will be assigned.

If the language is determined and not listed in `ok_languages` then the rule "UNWANTED_LANGUAGE_BODY" is triggered. That rule can be adjusted to give as many points as the users feels is apropriate. It defaults to add 2.8 to the "spam score". Adding

```
score UNWANTED_LANGUAGE_BODY 4.000
```

to local.cf would raise that score to four points.

Pyzor and Razor2

SpamAssassin's default behaviour is to use Pyzor and Razor2 if it is available. So in both Debian and Gentoo which don't change the defaults for `use_razor2` and `use_pyzor` installing the part the user wants to use is all the setup needed. On debian SpamAssassin requires razor2 so it is installed automatically, but if the user wants to use Pyzor as well the command:

```
marvin:/etc/spamassassin# apt-get install pyzor
```

Installs pyzor and it will be used by SpamAssassin. On Gentoo Linux

```
berbara root # emerge pyzor

berbara root # emerge razor
```

will install Pyzor and Razor2.

When using Pyzor or Razor2 an outbound tcp connection is made to the server hosting the spam list.

Therefore firewall setup should not be an issue unless the firewall setup is very strict.

Distributed Checksum Clearinghouse (DCC) is another spam collection database that checks around 130 million email per day. Debian has no official packages for DCC yet but Gentoo has. On Debain downloading and compilation from sources are required by the user. SpamAssassin will check for a working DCC installation and use that if found. DCC communicates by UDP to save as much bandwidth as possible. This makes firewall setup a little trickier, DCC requires incoming UDP communications on port 6277 and outgoing UDP on port 1023.

Testing SpamAssassins Setup

To test that everything is working with SpamAssassin and it's plugins SpamAssassin includes a text file (`sample-spam.txt`) which is guarantied to evaluate as spam. This file can be used in this manner:

```
marvin:~# locate sample-spam.txt
/usr/share/doc/spamassassin/examples/sample-spam.txt
/usr/share/doc/spamc/sample-spam.txt

marvin:~# spamc < /usr/share/doc/spamc/sample-spam.txt

[ Lots of output ]

Content analysis details:   (1004.2 points, 6.0 required)
 pts rule name              description
---- ---------------------- --------------------------------------------------
1000 GTUBE                  BODY: Generic Test for Unsolicited Bulk Email
 1.6 RAZOR2_CF_RANGE_51_100 BODY: Razor2 gives confidence between 51 and 100
                            [cf: 100]
 0.9 RAZOR2_CHECK           Listed in Razor2 (http://razor.sf.net/)
 0.3 PYZOR_CHECK            Listed in Pyzor (http://pyzor.sf.net/)
 1.4 DNS_FROM_RFCI_DSN      RBL: From: sender listed in dsn.rfc-ignorant.org

[ Lots of output ]
```

Under “Content analysis details” the different rules that have been identified can be checked and in the example we can see that both Pyzor and Razor2 is being used. If DCC was used a line similar to this will be displayed:

```
 1.8 DCC_CHECK              Listed in DCC (http://rhyolite.com/anti-spam/dcc/)
```

Combining Qmail and SpamAssassin

There are a few different ways to scan the mails going through our server. And the appropriate way will vary between each server/user scenario.

Here are a couple of things to consider when setting up spam filtering:

- Scanning incoming messages
    - Does all mail accounts need spam filtering?
    - Should spam be tagged or deleted?
- Scanning outgoing messages
    - Can you trust you users?

Setting up Qmail to scan all mail.

A simple way of making Qmail check all mail is to scan the mail before the mail is queued. All mail goes through `qmail-queue` to be queued and processed further. Have a look at appendix I you will see that `qmail-queue` is a single point of entry to the mail system. This is probably the simplest way to scan for spam.

First copy the original `qmail-queue` out of the way:

```
marvin:~# cd /var/qmail/bin
marvin:/var/qmail/bin# cp qmail-queue qmail-queue.orig
```

Then create a new `qmail-queue.new` file in your favorite text editor that contains this:

```
#!/bin/sh
/usr/bin/spamc | /var/qmail/bin/qmail-queue.orig
```

Note: *the hash bang(#!) at the beginning, that specifies what command the file should be run with. Be sure this points to a shell that exists.*

The new `qmail-queue.new` file will direct the incoming mail to the SpamAssassin client and the returned message from SpamAssassin will be piped to the original `qmail-queue.`

Now we have to make sure that our new `qmail-queue` has the same permissions as the old one:

```
marvin:/var/qmail/bin# ls -l qmail-queue.orig
-rwsr-xr-x    1 qmailq   qmail       15048 2004-03-17 00:57 qmail-queue.orig
marvin:/var/qmail/bin# chmod 4755 qmail-queue.new
marvin:/var/qmail/bin# chown qmailq.qmail qmail-queue.new
```

Now we can replace qmail-queue, like this:

```
marvin:/var/qmail/bin# mv qmail-queue.new qmail-queue
```

That is all that is needed to have all mails scanned by SpamAssassin. Note: There is a small disadvantage by doing it this way. If a local delivery is forwarded to another address (local or remote) the mail will be scanned twice, or even more times if there is multiple local forwards.

This is a very simple and none intrusive setup but it's not configurable. All mails will be scanned no matter what. It would be nice if we could use this setup for the smtp deamon to scan all incoming mails only, but because `qmail-smtpd` needs two-way communication with the client sending the

message this is not possible. We could patch `qmail-smtpd` to use use modified `qmail-queue` script but this also does provide a simple and powerful way of deciding what is checked for spam and what is not. A new approach is needed and in steps the `qmail-scanner` addon.

Qmail-scanner

`qmail-scanner` is an addon that enables a Qmail email server to scan all mail for certain characteristics.

Support for qmail-scanner needs the qmail package to be patched, but from the installation section you will see that in both Gentoo and Debian Linux these patches are applied by default. They are so common that if it was not for D.J. Bernstein's license they would be part of the Qmail package.

Dependecies tnef for special ms attachments

Run the configure script:

```
marvin:~/build/qmail-scanner-1.22# ./configure
Building Qmail-Scanner 1.22...

This script will search your system for the virus scanners it knows about, and will
ensure that all external programs qmail-scanner-queue.pl uses are explicitly pathed
for performance reasons.

It will then generate qmail-scanner-queue.pl - it is up to you to install it
correctly.

Continue? ([Y]/N)

/usr/bin/uudecode works as expected on system...
Found tnef on your system! That means we'll be able to decode stupid M$
attachments :-)
The following binaries and scanners were found on your system:
mimeunpacker=/usr/bin/reformime
uudecode=/usr/bin/uudecode
unzip=/usr/bin/unzip
tnef=/usr/bin/tnef

Content/Virus Scanners installed on your System

clamscan=/usr/bin/clamscan
fast_spamassassin=/usr/bin/spamc

Qmail-Scanner details.

Log-details=0
fix-mime=2
ignore-eol-check=0
debug=1
notify=psender,nmlvadm
redundant-scanning=no
virus-admin=root@marvin
local-domains='marvin'
silent-
viruses='klez','bugbear','hybris','yaha','braid','nimda','tanatos','sobig','winevar','
palyh','fizzer','gibe','cailont','lovelorn','swen','dumaru','sober','hawawi','holar-
i','mimail','poffer','bagle','worm.galil','mydoom','worm.sco','tanx','novarg','\@mm'
scanners="clamscan_scanner","fast_spamassassin"

If that looks correct, I will now generate qmail-scanner-queue.pl
for your system...
Continue? ([Y]/N)
```

Sources

Qmail.org http://www.qmail.org/

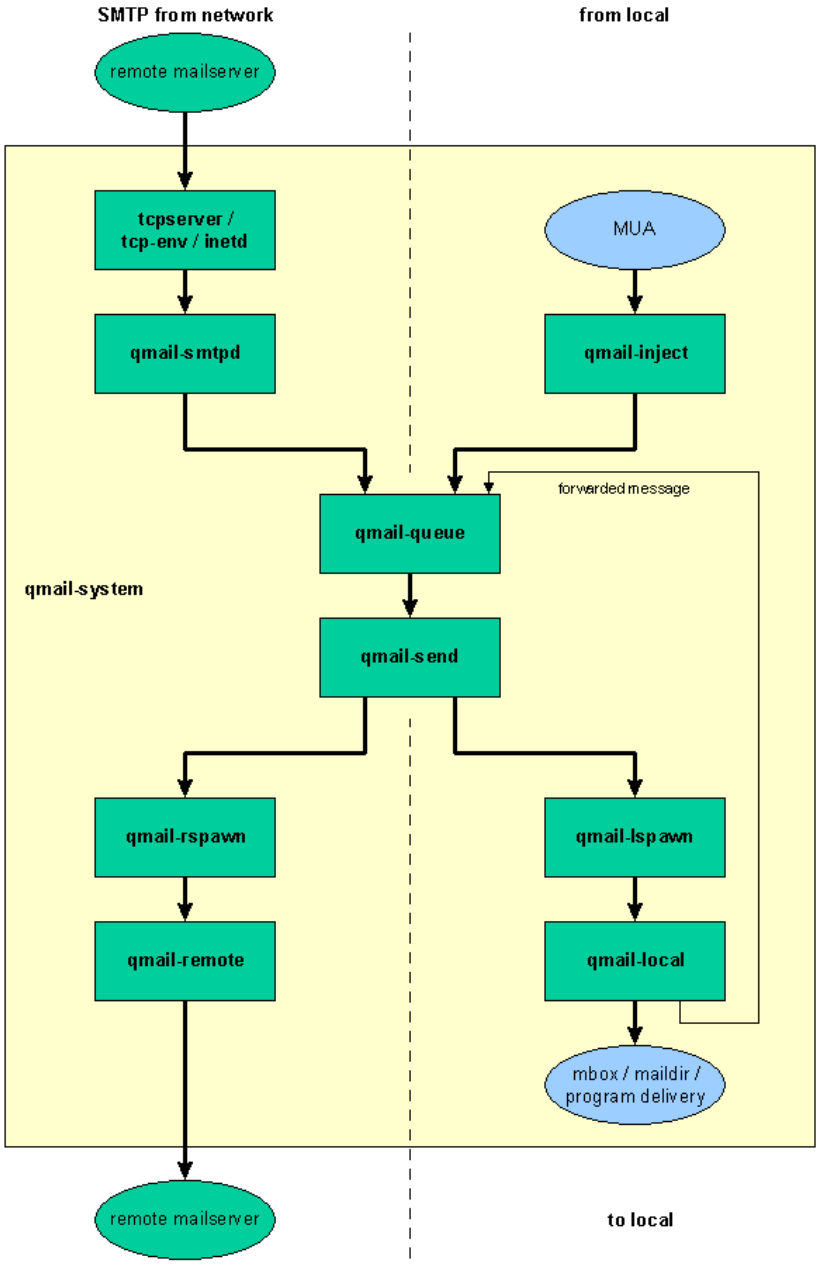
Life With Qmail http://www.lifewithqmail.org/

D. J. Bernstein's home pages http://cr.yp.to/

Various Linux man pages (spamassassin, Mail::SpamAssassin, Mail::SpamAssassin::Conf, qmail, qmail-send, qmail-queue, ...)

Appendix I

The Big Qmail Picture (http://www.nrg4u.com/)



```
      ****** FINAL TEST ******

Please log into an unpriviledged account and run
/var/qmail/bin/qmail-scanner-queue.pl -g

If you see the error "Can't do setuid", or "Permission denied", then
refer to the FAQ.

(e.g.  "setuidgid qmaild /var/qmail/bin/qmail-scanner-queue.pl -g")


That's it! To report success:

   % (echo 'First M. Last'; cat SYSDEF)|mail jhaar-s4vstats@crom.trimble.co.nz
Replace First M. Last with your name.

marvin:~/build/qmail-scanner-1.22# apt-get install perl-suid
```