A QMail Based Mail Server

E-Mail from the Weirdo

Deciding on a mail server is no easy task. Should you follow the herd, opting for Sendmail simply because it's the most popular? This is not the option an admin would choose if she was intent on providing a simple configuration that would not giver her fellow admins headaches. There are alternative Mail Transfer Agents – QMail is secure and easier to operate.

BY FRANK BERBIG

The sheer bulk of the mail messages both from home users and enterprises on the Internet continues to grow a day by day, and ever since the ILOVEYOU virus hit, both users and admins are aware of the danger and e-mail based attack on a local network can cause. Thus, the availability and protection of mail systems is becoming increasingly important.

Having said that, to be secure the system needs to be straightforward, easy to understand and manageable. Being the only person in the whole enterprise that understands the Sendmail configuration may keep you in a job, but what happens if you get sick? Basing the critical e-mail infrastructure of an enterprise or organization on a Mail Transfer Agent (MTA) that does not require the admin to plough through manuals the size of bricks saves time, nerves and is future-proof. And that's why we will be looking into migrating a standard distribution from Sendmail to QMail in this issue.

What is QMail?

QMail is a lightweight, quick, and reliable MTA for UNIX-style operating systems authored by Dan J. Bernstein (DJB), a mathematics professor at the University of Illinois in Chicago. It was designed as a replacement for Sendmail, which has had its share of security holes and other scares in the past.

Version 1.03 was released in June 1998, and the basic system has remained unchanged ever since. The refusal of the author to continue developing the system, accompanied by a fair degree of stubbornness, are often quoted as points of criticism on QMail. On the other hand, the fact that the system has not require any bug fixes for four years says a lot about the quality of the code

Sendmail's unintuitive configuration files give birth to the myth that installing the mail server is something akin to wizardry. Although QMail's installation may be somewhat more Spartan than alternative MTAs, you do not need to be a guru to make things work. What you do need is sound working knowledge of UNIX-style operating systems, of what a mail server really does [1], and a fair amount of patience; the latter, mainly due to the fact that installing QMail is not a simple matter of running a script.

The central QMail website (that is *http://cr.yp.to/qmail.html*, *http://www.qmail.org/* and *http://www.lifewithqmail.org/*) is a good place to start if you're looking for software packages and documentation, or if you need help solving individual problems or complying with specific demands.

Mbox versus Maildir

But before we dive headlong into the QMail adventure, you should acquaint yourself with one or two peculiarities of this MTA. These include the format that QMail uses for storing messages. Messages are placed in the users' home directories by default and not in the spool directory */var/spool/mail/username*.

One of the main reasons for this is the fact that QMail uses the Maildir format for storing mail by default. The integrated Mail Delivery Agent (MDA) does not follow the typical pattern of adding mail to the user's spool file (Mbox format), but stores each message is a separate file instead.

This type of storage resolves the issue of concurrent access to a file that can occur when a user wants to collect her mail, and the MTA attempts to store a new message at the same time. Additionally, there are no issues with file sizes. If every message is stored in a single file, the file may easily grow to several hundred MB in size given sufficient mail throughput.

Every QMail mail directory and every subfolder contains three directories: *cur*, *new*, and *tmp*. *cur* is used for storing messages that the user has already viewed or read, whereas any unread new messages are stored in the directory

Qmail Installation

Qmail is compiled individually from the source code. To do this you require the *qmail-1.03.tgz* [4] *ucspi-tcp-0.88.tar.gz* [5] packages.

This will initially create a local mail server without any access to exterior networks, that is without an SMTP service on Port 25 (and of course without POP or IMAP services). First, unpack the archives in a temporary directory (for example */tmp*) and then change to the Qmail source code directory created in this step:

```
cd /tmp
tar xzvf qmail-1.03.tgz
tar xzvf ucspi-tcp-0.88.tar.gz
cd /tmp/qmail-1.03
```

The directory should contain the *conf-qmail* that specifies the installation directly for the MTA. If you're happy with the default, */var/qmail*, you can simply call *mkdir* to create the directory.

If you prefer to install Qmail in a different directory, you need to edit the entry in *conf-qmail* to reflect this. Be aware that this setting will also influence packages that you may install later.

To create the users required by Qmail, logon as the *root* user and copy the *INSTALL.ids* file (to *ids.sh*, for example), remove the lines that precede the first instance of *# groupadd nofiles*, and the lines at the end of the file including and following *FreeBSD 2.2*. Now remove the comment tag # from the remaining nine lines, add *#!/bin/sh* as the first line, make the file executable, and finally launch it:

```
cp INSTALL.ids ids.sh
vi ids.sh
chmod 755 ids.sh
./ids.sh
```

The script thus made will create the required users and groups. To compile and install the MTA now type the following command:

```
make setup check
```

To configure a server that will be handing mail for a specific domain, ensure that the source code directory is your working directory and type the following command (the abbreviation tld means "Top Level Domain", for example *co.uk*):

```
./config-fast my.domain.tld
```

If you want the server to handle local mail only, as described above, *localhost* is the only argument you will need.

The *ucspi-tcp* package installs tools (which are principally independent of Qmail) that allow you to create a TCP client/server application. After typing the following to install:

```
cd /tmp/ucspi-tcp-0.88
make
make setup check
```

the */usr/local/bin* should contain a collection of executables: *addcr*, *argvo*, *date@*, *delcr finger@*, *fixcrio*, *http@*, *mconnect*, *mconnect-io*, *rblsmtpd*, *recordio*, *tcpcat*, *tcpclient*, *tcprules*, *tcprulescheck*, *tcpserver*, and *who@*. We will not be looking into these programs as they are beyond the scope of this article.

The */var/qmail/boot* contains a collection of scripts for launching Qmail. The following script shows an extremely simple example:

```
#!/bin/sh
exec env -
PATH="/var/qmail/bin:$PATH"
qmail-start ➥
"`cat
/var/qmail/control/defaultdeliver
y`" splogger qmail
```

When stored as */var/qmail/rc*, the script specifies that the default delivery type – Maildir – will be read from the file */var/qmail/control/defaultdelivery*. To create this file log on as *root* and type:

```
echo ./Maildir/ > ➥
/var/qmail/control/➥
defaultdelivery
```

After making */var/qmail/rc* executable, you can launch the mail server by typing:

```
/var/qmail/rc &
```

Before doing so, make sure to stop and disable Sendmail (or the MTA used by your system). Additionally, you might like to remove any rights assigned to the Sendmail program:

```
/etc/init.d/sendmail stop
mv /pfad/zu/sendmail ➥
/pfad/zu/sendmail.old
chmod 0 /pfad/zu/sendmail.old
```

To handle applications that assume Sendmail as their mail program, ensuring that they continue to work after migrating, Qmail provides a Sendmail wrapper. The wrapper responds just like Sendmail, but uses Qmail as its MTA:

```
ln -s /var/qmail/bin/sendmail ➥
/path/to/sendmail
```

Finally, a few mail aliases need to be created:

```
echo admin > ➥
/var/qmail/alias/.qmail-root
echo admin > ➥
/var/qmail/alias/.qmail-
postmaster
ln -s /var/qmail/alias/.qmail-
root ➥
/var/qmail/alias/.qmail-mailer-
daemon
chmod 644
/var/qmail/alias/.qmail-root ➥
/var/qmail/alias/.qmail-
postmaster
```

admin, the user responsible for administrating Qmail will normally already have an account on the mail server. If not, you will need to create the account.

Alias file names can be interpreted as follows: the section following the hyphen yields the name that the alias will answer to (e.g. *.qmail-root* yields *root*). If Qmail is required to deliver a message to the superuser, it searches for the alias, reads the file content and attempts to deliver the message to the user named by the file: admin in this case. Following this, Qmail attempts to follow the mail delivery instructions found in the *.qmail* file in the user's home directory. If the alias file quotes *root* as the users, the message cannot be delivered, as Qmail refuses to deliver messages directly to the superuser.

called *new*. QMail uses *tmp* to temporarily store new mail before delivery.

All for One

The fact that QMail impresses with its security and speed, instead of attracting unwanted attention to its security holes, is partly due to the fact that the mail server is not a monolithic program in contrast to Sendmail, which stores all its functions as a single binary. Unlike QMail, which comprises of several smaller program, each of which fulfills a specific task and where each of these tasks runs as a different user with a seperate user account.

Parallel mail delivery accounts for the speed of the program. Users report [2] 450,000 messages handled on a single day by a Pentium 90 with 64 MB of RAM!

QMail's security also benefits from its architecture: when the MTA excepts a message, it also guarantees delivery.

The software achieves this by first queuing the incoming message and not deleting the message from the queue until it has been delivered to the user. This drastically reduces the risk of message loss, which is an obvious benefit in any mail system.

Installation

The following steps assume a fully functional Linux system, preferably a recent release. Sendmail may be preinstalled and even running (the case of Red Hat 8.0, for example), however, more distributors are starting to use alternative MTAs (although they tend not to opt for QMail).

Within the framework of the installation steps described in the "QMail Installation" inset, the installation described in the source code – an excerpt from the special QMail license – QMail replaces Sendmail. Although the installation procedure should be no different for various distributions, the steps required to disable the preinstalled MTA will depend on the local scenario.

The QMail installation procedure does not require a nameserver, however, this article assumes a functional and accessible nameserver. The Howto at [3] provides good overview of the "Nameserver and DNS" topic.

Configuration, but where?

Outputting the basic configuration script *config-fast* (see installation inset) indicates that the files in the */var/qmail/control* contain the central QMail setup parameters. A separate file is used for each configuration parameter.

The most important configuration files are *defaultdelivery*, *defaultdomain*, *locals*, *me*, *rcpthosts*, and *virtual domains*. *defaultdelivery* specifies the standard delivery method (Maildir or Mbox); *locals* specifies the domains that this mail server manages.

The manpage for *qmail-control(5)* has a list of control files. If */var/qmail/man* is not in the search path for the *man* command, you can enter the following:

```
groff -mandoc -Tlatin1 ➥
/var/qmail/➥
man/man5/qmail-control.5 | less
```

to view the manpage.

The help page indicates the name, the default setting, and the program that uses the setting. The manpages for the individual programs contain more details and also describe the appropriate control files that you will come across.

Delivery Paths

QMail parses incoming messages for target addresses and checks their domain section to find out whether the message is intended for the local server. In this case, the program reads the name section up to the @. If the user exists, the MDA attempts to write the message to the user's home directory.

If there is no local user, QMail checks for a matching alias, that is, a file with the name *.qmail*-namesection. If an alias exists, QMail forwards the message to the user named in the file.

In case of alias files, note that the name section does not contain any periods (.). If periods are needed for some reason, you need to replace them with colons. The alias file *.qmail-frank:berbig* thus contains an alias for *frank.berbig*.

After discovering the user, the ensuing mail handling procedure depends on the content of the *.qmail* file in each of the user's home directory. The so-called "dot qmail file" offers the following possibilities:

- Adding commentaries: *# Commentary*
- Delivery to Maildir: *./Maildir/*
- Delivery to an Mbox file: *./mbox*
- Handling by a third-party program *|/path/to/program* arguments
- Forwarding: *me@host.tld*

If you want your mail server to speak IMAP

It should not be necessary to run SMTP, POP and IMAP servers with their various tasks on a single server, and is definitely not the rule for larger mail providers. However, if you really do want to use one physical sever for all of these jobs, you might like to look into the tried and trusted Courier IMAP Daemon (available at *http://www.inter7.com/courierimap/*), while continuing to use QMail as your SMTP (and possibly POP3) server.

During the installation steps ensure that you neither unpack nor compile the server as *root*. The best way to go here, is to create a special account called *imapd* and use it to perform the install steps. The following commands unpack, compile and install Courier IMAP:

```
tar xzvf courier-imap-version.tar.gz
cd courier-imap-version
./configure
make
make check
su
make install
make install-configure
```

The subdirectory *etc* in the Courier IMAP directory, */usr/lib/courier-imap/*, now contains the central configuration file, *imapd*. Typically, no changes are required. A matching start script is stored as *imapd.rc* in *libexec*.

The only modification to the basic install would be to replace *authmodulelist* with *authshadow* in the *authdaemonrc* file, thus ensuring that the system password file */etc/shadow* is used for authentication purposes. If you need to authenticate your IMAP users against a different source, such as a MySQL database or PAM, you should consult the documentation for additional details.

Unfortunately, there is a trap waiting for the user here: although QMail will automatically create a file called *mbox* in the user's home directory as soon as a message arrives, when *./mbox* is specified in *~/.qmail*. This is not the case if a maildir is specified, the user will first need to run */var/qmail/bin/maildirmake ~/Maildir* to create the appropriate maildir in her home directory, to allow the MDA to deliver.

Talk to me!

If you launch the QMail server as described in the "QMail Installation" inset, and then enter *telnet localhost smtp* in an attempt to talk to the server, you might be surprised by the *Connection refused* response, although QMail is obviously alive, as the message dispatched using *mutt* and appearing in the */var/log/mail* logfile, shows.

Before it can talk to the outside world, the MTA needs a few more programs, that permit POP, SMTP, and IMAP access.

POP, the Post Office Protocol, allows mail clients to fetch mail from remote mail servers without the local computer running an MTA and MDA that permit external access. The messages are sent to the requesting client after it has authenticated, and then (typically) removed from the server. QMail has its own POP daemon which is set up by modifying */etc/inetd.conf*:

```
pop3 stream tcp nowait root ↩
/var/qmail/bin/qmail-popup ↩
qmail-popup your_hostname ↩
/bin/checkpassword ↩
/var/qmail/bin/qmail-pop3d ↩
Maildir
```

as follows and restarting *inetd*:

```
/etc/init.d/inetd restart
```

When modifying the inet daemon configuration file, ensure that only one entry exists for each protocol (*pop3* in this case). A similar configuration entry allows external access to QMail as an SMTP server using the Simple Mail Transfer Protocol:

```
smtp stream tcp nowait qmaild ↩
/var/qmail/bin/tcp-env
tcp-env /var/qmail/bin/↩
qmail-smtpd
```

IMAP, the Internet Message Access Protocol provides extended functionality, allowing clients to manage messages on the server, instead of having to download them to the local machine, as is the case for POP. In contrast to POP, however, QMail does not provide a tool of its own. However, adding the Courier IMAP Server can add this facility to the mail server (see boxout "If you want your mail server to speak IMAP").

In Production

If you operate a mail server and often need to create new users, you might not feel inclined to create the appropriate maildir and *~/.qmail* file manually, every time. Fortunately, most distributions provide a *skel* directory below */etc* that allows you to create a kind of template for user directories. The files and directories you create in *skel* are copied, along with appropriate permissions, user and group ownerships, when a new account is created. The following commands add QMail specific user files to the */etc/skel* template:

```
/var/qmail/bin/maildirmake ↩
/etc/skel/Maildir
echo "./Maildir/" > /etc/skel/↩
.qmail
```

In case of complaints about messages not being delivered, it makes sense to take a look at the log files. Unfortunately, QMail writes numerous entries that are definitely irrelevant to troubleshooting. To filter the relevant details, you first need to know what you are looking for: in the log file QMail errors contain the

Log Stats

Dan Bernstein's Tools from the *qmailanalog* package, or the *Isoqlog* package provided by the QMail community are useful for creating meaningful statistics from log files. Qmailanalog is available from [6]. To install, perform the following steps:

```
make
make setup check
```

The following command should create some input for the tools

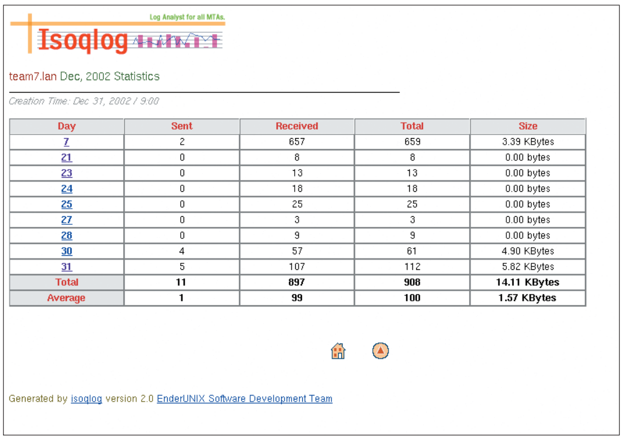
```
cat /var/log/mail | awk '{$1="";↩
$2="";$3="";$4="";$5="";print}'↩
> /tmp/qmailanalog
```

This feeds the content of the */tmp/qmailanalog* file to the *matchup* command, and its output is then parsed by one of the "Z tools" – depending on what kind of statistics you need. The directory used for storing the Qmailanalog programs depends on your installation:

```
cat /tmp/qmailanalog | ↩
/path/to/matchup | ↩
/path/to/zoverall > resultfile
```

A short script that creates the result file and stores it somewhere where your web server can access it, and regular calls to the script by cronjobs, will ensure that your statistics remain up to date and easy to read. However, the quality and usefulness of these statistics is a source of endless debate.

The limited scope of this article prohibits a more detailed description of Isoqlog. The installation for this package, which is available from [7], follows the usual pattern *./configure*, *make*, and *make install*; the *isoqlog.conf* file in the installation subdirectory *etc* is used to configure the tool. All this effort is rewarded by linked HTML pages, preferably in a directory accessible to your web server.



team7.lan Dec, 2002 Statistics

Creation Time: Dec 31, 2002 / 9:00

Day	Sent	Received	Total	Size
7	2	657	659	3.39 KBytes
21	0	8	8	0.00 bytes
23	0	13	13	0.00 bytes
24	0	18	18	0.00 bytes
25	0	25	25	0.00 bytes
27	0	3	3	0.00 bytes
28	0	9	9	0.00 bytes
30	4	57	61	4.90 KBytes
31	5	107	112	5.82 KBytes
Total	**11**	**897**	**908**	**14.11 KBytes**
Average	**1**	**99**	**100**	**1.57 KBytes**

Generated by isoqlog version 2.0 EnderUNIX Software Development Team following pattern *(#[0-9].[0-9].[0-9])*. Thus, the following syntax displays the errors:

```
egrep '\(#[0-9]\.[0-9]\.[0-9]➥
\)' /var/log/mail
```

Appending | *tail -1* will display only the latest errors to have occurred. This output is a lot easier to analyze. The "Log Stats" boxout shows how to compile regular and up to date statistics from the log file.

When you finally have the QMail server up and running errors are fairly uncommon. However, QMail is very sensitive to incorrect user *Maildir* permissions. To avoid unexpected vacuums in various mailboxes careful admins might like to run a Perl script to check permissions. *checkhomeownership* is available at [8].

The disadvantage of this is that it simply checks the home directories of all the users listed in */etc/passwd*, and some of these will not have a maildir of their own, being system accounts.

Being bombarded with error messages for these accounts each time you run the script, does not make much sense, so you might like to add these users to a kind of negative list. The *checkhomeownership.sh* [9] shell script eliminates these users from the Perl script's output, and sends the results to the admin user.

The QMail Queue

Messages that QMail will be handling are stored in the mail queue below the */var/qmail/queue* directory. Although :

```
find /var/qmail/queue/ -type f
```

will list the files in the mail queue, you might prefer a more convenient approach. QMail provides the *qmail-qstat* viewing tool; you can display information on the mails sitting in the queue as follows:

```
/var/qmail/bin/qmail-qread
```

qmail-send will attempt to reschedule delivery of these messages. If too many delivery errors occur, QMail issues an error message. However, if you discover that mail is getting stuck in the queue rather than being removed automatically, you can use *qmHandle* [10] to delete the queue

Dialup Connections

If the local network, or mail server, only has a dialup connection via modem or ISDN, you cannot deliver external mail directly. Instead, it makes sense to collect the messages until an Internet connection is re-established.

QMail handles this issue by using a special account (such as *outgoing*, for example) to store mail that cannot be delivered to a (local) user. To use the account, add the following line:

```
:alias-outgoing
```

to the end of the control file *virtual domains*, and create an alias file called *.qmail-outgoing-default* containing *out going* in the *alias* directory.

Mail stored on the *outgoing* account needs to be sent to an external mail server (hosted by a provider, for example), when an Internet connection is established.

The serialmail [11] package takes care of this for dialup connections. Before using *make* and *make setup check* to install, you may need to modify the *conf-home* and *conf-qmail* files; the accompanying *INSTALL* file contains additional information.

Following the installation steps, the programs, including *maildirsmtp*, will be located in the *conf-home/bin* directory. The following command:

```
/path/to/maildirsmtp ➥
/home/outgoing/Maildir➥
/  "alias-outgoing" ➥
external_mailserver local_server
```

will dispatch any messages below */home/outgoing/Maildir/new* to the external mail server. The *alias-outgoing* entries added to the mail addresses by QMail are eliminated during this process. If an empty parameter is used on the command line instead of "*alias-outgoing*", the replace operation will not occur.

If the mail server accessed by this operation will not accept messages until the user has cleared her POP account ("SMTP after POP"), you will need to call *fetchmail* before calling *maildirsmtp*.

Conclusion

The effort involved in installing QMail on a Linux system is definitely non-trivial. However, the server will reward you with a high level of stability and infrequent updates.

The fact that the user community is extremely active speaks in favor of using QMail and ensures a steady supply of documentation, additions and patches. This is also why QMail is not only one of the safest, but also one of the most complete MTAs.

A word of caution: the installation described here is not suitable for a production mail server connected to the Internet, as it does not provide any protection against relaying, spam or viruses.

Adding these facilities aswell as mailing list management with QMail are discussed in part 2 of this article in a following issue of Linux Magazine. ■

INFO

[1] Marc André Selig: "E-Mail entwirrt", Linux-Magazin 04/2002, S. 32 ff.

[2] *http://cr.yp.to/qmail/users.html*

[3] *http://www.linuxhaven.de/dlhp/HOWTO/DE-DNS-HOWTO.html*

[4] *ftp://ftp.eu.uu.net/pub/unix/mail/qmail/qmail-1.03.tar.gz*

[5] *http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz*

[6] *http://cr.yp.to/software/qmailanalog-0.70.tar.gz*

[7] *http://www.enderunix.org/isoqlog/*

[8] *http://qmail.qbiz.de/checkhomeownership*

[9] *http://www.linux-magazin.de/Service/Listings/2003/05/QMail/checkhomeownership.sh*

[10] *http://www.io.com/~mick/soft/qmHandle-0.5.1.tar.gz*

[11] *http://cr.yp.to/software/serialmail-0.75.tar.gz*

THE AUTHOR

Frank Berbig is a freelance web developer based in South Germany. Frank uses PHP, Java, and various databases, such as MySQL, PostgreSQL, and Oracle for his web development work, which involves frequent – and welcome – contact with Linux servers.