Exim Practical

(based on materials from Brian Candler)

Objectives

Part 1 is building and installing Exim.

- Install Exim from ports
- Replace Sendmail with Exim

Part 2 is running basic tests. You don’t need to modify the Exim configuration to do this.

- Test a standard installation and default configuration
- Inspect and manage the mail queue
- Check relay control
- Process log data

Part 3 involves some simple modification of the runtime configuration.

- Modify the runtime configuration to send undeliverable mail to postmaster

Part 4 sets up your host as a mail relay

- Allow relaying from another host
- Allow relaying to another domain

Common mistakes

In past workshops, these are the most common mistakes that have been made:

- *Doing everything as root.* You only need to be root to install Exim and change its configuration. Otherwise, you should do everything under your normal login. In the sample commands, the command prompt is shown as **#** for commands that must be run as root, and **$** otherwise.

  In particular, running email tests as root is a bad idea, because root has privileges. You want to test that Exim is working when an ordinary, unprivileged user calls it.

- *Forgetting the dot that terminates a message.* When you type a message directly into Exim, it needs a line containing just a dot to terminate it. Until you type that line, all input is taken as part of the message.

- *Adding dots to email domains.* You should now have got used to inserting trailing dots in fully qualified domains in DNS zones. Unfortunately, in email configurations, trailing dots are *not* used and will cause problems if you use them.

1. Installing Exim

As with most software under FreeBSD, you have three choices on how to install it:

1. Install a binary package from the CD-ROM or an FTP server
2. Build from the ports collection
3. Download the source code and compile and install it yourself following the author's instructions.

Here we are going to install Exim from ports. It's a straightforward process, easier than compiling it yourself. If you bring your ports collection up to date using cvsup then you will be building the latest version of Exim (whereas the binary package on the CD-ROM may be an older version).

Build and install from ports

Proceed as follows:

```
# cd /usr/ports/mail/exim
# make
... a lot of stuff is displayed ...
# make install
... some more stuff is displayed ...
# rehash      (Needed if you are using the C shell)
```

You should end up with the Exim program in **/usr/local/sbin/exim** and a default configuration file in **/usr/local/etc/exim/configure**. If you successfully completed an update of your Ports collection previously, then Exim version 4.51 should be installed (June 12, 2005).

(ASIDE: There are optional extra features you can build into exim when compiling it. For example, "make WITH_MYSQL=yes" would build exim with MySQL support, so that mailboxes could be looked up in a MySQL database. However, the defaults in the ports collection are fine for what we need. Look at the comments in `/usr/ports/mail/exim/Makefile` to see the options available).

If a **sendmail** daemon is running, kill it off. Use **ps** with **grep** to find it. (Modern versions of sendmail may have two or more processes running)

```
# ps auxw | grep sendmail
# /etc/rc.d/sendmail stop
# vi /etc/rc.conf
... set this value:
sendmail_enable="NONE"
```

Note that the FreeBSD port builds exim to run using user "mailnull" and group "mail". You should now arrange for your personal (non-root) account to be in the **mail** group so that you can be an administrator for Exim.

Remember our `pw usermod` discussion earlier in the day. If you use the "-G" option you need to specify all groups a user belongs to as well as the new group - otherwise the user will be removed from any groups not specified.

```
# groups username
... Note the groups your user belongs to (such as wheel)
# pw usermod username -G mail,wheel
... you don't need to specify the username group
# grep mail /etc/group           (To check)
```

Check the documentation

Before moving on, make sure you can access the Exim documentation. It is available as a plain-text file which can be read or searched using the normal Unix tools or a text editor:

```
$ less /usr/local/share/doc/exim/spec.txt
```

(There is also a separate port available, `mail/exim-doc-html`, which installs the Exim documentation in HTML format for reading with a web browser. Or it's available on-line at http://www.exim.org/)

Test that Exim has been installed by running:

```
$ exim -bV
```

which should tell you Exim’s version number and some other information about which features are included.

Replace Sendmail with Exim

All the MUAs call **/usr/sbin/sendmail** to pass messages to the MTA. We want them to call exim instead of sendmail, but leave sendmail installed so that if there is any existing mail in sendmail’s queue, it can be flushed later.

FreeBSD lets us choose a new MTA by setting values in the `/etc/mail/mailer.conf` file. We redirect the programs ’sendmail’, ’send-mail’ and ’mailq’ to point to exim.

```
# cd /etc/mail
# vi mailer.conf
... change these three lines:
sendmail        /usr/local/sbin/exim
send-mail       /usr/local/sbin/exim
mailq           /usr/local/sbin/exim
... you can leave newaliases, hoststat and purgestat unchanged
```

Now try that basic test again, but this time using the standard path name:

```
$ /usr/sbin/sendmail -bV
```

You should get the same output as before, which shows that Exim is now being used instead of Sendmail.

2. Testing Exim

If you are doing a real installation on a live system, you might want to work on the configuration and do lots of testing before removing Sendmail and replacing it with Exim.

Test the standard installation and configuration

Make sure you substitute a real local user name for *localuser* in what follows. Remember, you should not be root when running these tests.

First, check what Exim will do with a local address:

```
$ exim -bt localuser
```

This tests the delivery routing for a local account. See what output you get.

Try with a non-existent local user and see what happens:

```
$ exim -bt junkjunkjunk
```

Try something that is in **/etc/aliases**:

```
$ exim -bt postmaster
```

Exim will not normally deliver mail to a *root* mailbox (for security reasons) so what people usually do is to make *root* an alias for the sysadmin. In FreeBSD, all the default aliases point to *root*. Therefore, you should add a new alias to **/etc/aliases** (you will need to be root to do this).

Find the following line:

```
# root: me@my.domain
```

and change it (remembering to remove the initial ’#’) to:

```
root: localuser
```

Now try this again:

```
$ exim -bt postmaster
```

Now we are going to try a real local delivery. You can pass a message directly to Exim without using an MUA:

```
$ exim -v -odf localuser
This is a test message.
.
```

Note: the message is terminated by a line that just contains a dot. Be sure to type it! (Alternatively, you can send ‘‘end of file’’ by pressing CTRL-D.)

The `-v` option turns on user verification output, which shows you copies of Exim’s log lines.

The `-odf` option requests ‘foreground’ delivery, which means that the **exim** command won’t return until the delivery is complete. (This avoids your shell prompt getting mixed up with Exim’s output.)

Check what is in Exim’s logs:

```
$ cat /var/log/exim/mainlog
$ cat /var/log/exim/paniclog
```

The panic log should normally be empty, and if nothing has ever been written to it, it will not even exist. *Tip*: On a live system it is helpful to set up a **cron** job that mails you a warning if it ever finds a non-empty panic log.

If you get a permission error, make sure that your username is in the ’mail’ group, then logout and login again to become a member of that group.

If the delivery succeeded, you should see two lines in the main log, one containing `<=` for the message arriving, and one containing `=>` for the delivery.

Now go check the local user’s mailbox:

```
$ ls -l /var/mail/localuser
$ cat /var/mail/localuser
```

If the delivery didn’t succeed, you need to find out why. If the information in the log doesn’t help, you can try the delivery again, with debugging turned on:

```
$ exim -d -odf localuser
<there will be output from Exim here>
This is another test message.
.
```

The `-d` option turns on debugging, which gives a lot more information than `-v`. You need to be an Exim administrator to use `-d`. If you get a ‘Permission denied’ error, check that you are a member of the "mail" group.

If you are logged on as *localuser*, you can use the *mail* command to read the mail in the usual way. You could also try sending a message from the *mail* command.

The next thing is to test whether Exim can send to a remote host. The speed of this may vary, depending on the state of the network connection. In what follows, replace *user@remote.host* with a remote email address that you use and can access.

First, check that Exim can route to the address:

```
$ exim -bt user@remote.host
```

Here’s a sample of what you should get back (address used was herveyallen@fastmail.fm):

```
herveyallen@fastmail.fm
  router = dnslookup, transport = remote_smtp
  host in1.smtp.messagingengine.com [66.111.4.73]   MX=10
  host in1.smtp.messagingengine.com [66.111.4.71]   MX=10
  host in1.smtp.messagingengine.com [66.111.4.72]   MX=10
  host in1.smtp.messagingengine.com [66.111.4.70]   MX=10
  host in2.smtp.messagingengine.com [66.139.75.100] MX=20
```

Now send a message to the remote address:

```
$ exim -v -odf user@remote.host
This is a test message.
.
```

This time, the `-v` option causes Exim to display the SMTP dialogue as well as the log lines. If you can, check that the message arrived safely. If there are problems, see if you can figure out what went wrong and why.

Here’s a sample of the end of on-screen output from a successful send:

```
  SMTP>> MAIL FROM: SIZE=1346
  SMTP>> RCPT TO:
  SMTP>> DATA
  SMTP<< 250 Ok
  SMTP<< 250 Ok
  SMTP<< 354 End data with .
  SMTP>> writing message and terminating "."
  SMTP<< 250 Ok: queued as 1751E929DFD
  SMTP>> QUIT
LOG: MAIN
  => herveyallen@fastmail.fm R=dnslookup T=remote_smtp H=in1.smtp.messagingengine.com [66.111.4.70]
LOG: MAIN
  Completed
```

You won’t be able to receive messages from a remote host until you start the Exim daemon (you need to be root to do this):

```
# exim -bd -q30m
```

The `-bd` option causes the daemon to listen for incoming SMTP calls, and the `-q30m` option causes it to start a queue runner process every 30 minutes. On a live system, starting the daemon should happen automatically on a reboot, by putting the following entry in `/etc/rc.conf` :

```
exim_enable="YES"
```

Once you’ve done this, you can use `/usr/local/etc/rc.d/exim.sh {start|stop|status}` as usual.

Use telnet to check that the daemon is accepting SMTP calls:

```
$ telnet localhost 25
```

You should see an Exim greeting message. You can type "quit" to exit.

The next two tests will work as long as each machine has a fully resolvable dns entry as well as an MX record. Your instructor will let you know if this is not the case.

Now check that a remote host can send a message to your host, and see how Exim logs what happens. If that succeeds, you have a working basic installation correctly installed.

Try sending to an invalid address from a remote host, and see what error message you get, and how Exim logs this case. Look in both **mainlog** and **rejectlog**.

Queue management tests

There are several command line options for doing things to messages.

To put a message on the queue without its being delivered, run

```
$ exim -odq address1 address2 ...
Test message.
.
```

The message stays on the queue until a queue runner process notices it.

List the messages on the queue:

```
$ exim -bp
```

Do a manual queue run, with minimal verification output:

```
$ exim -v -q
```

(Without `-v` you won’t see any output at all on the terminal, but there will be entries in the log.)

Checking relay control

To demonstrate that Exim will relay by default via the loopback interface, try the following sequence of SMTP commands. Wait for Exim to respond to each command before typing the next one. Substitute the number of your pc for *nn*.

```
$ telnet 127.0.0.1 25
ehlo localhost
mail from:<localuser@pcnn.pacnog.schoolfj>
rcpt to:<localuser@pcnn.pacnog.school.fj>
rcpt to:<user@some.remote.domain>
```

You should get an OK response to all the SMTP commands. Type ‘quit’ to end the SMTP session without actually sending a message.

Now try the same thing, but use your host’s IP address instead of 127.0.0.1.

```
$ telnet 202.62.122.nnn 25
ehlo localhost
mail from:<localuser@pcnn.pacnog.schoolfj>
rcpt to:<localuser@pcnn.pacnog.schoolfj>
rcpt to:<user@some.remote.domain>
```

In this case, you should get the error message

```
550 relay not permitted
```

for the second RCPT command, which is the one that is trying to relay. The first RCPT command should be accepted, because it specifies a local delivery. You could also try telnetting from an external host and running the same check.

Processing log data

Run **exigrep** to extract all information about a certain message, or a certain user's messages, or messages for a certain domain. For example:

```
$ exigrep localuser /var/log/exim/mainlog
```

That extracts all the log information for all messages that have any log line containing '*localuser*'. It's a Perl pattern match, so you can use Perl regular expressions.

To extract simple statistics from a log, run

```
$ eximstats /var/log/exim/mainlog | more
```

There are options for selecting which bits you don't want. Details are in the manual. If you have time, experiment with the options for outputting the statistics as HTML.

3. Changing the configuration

To change Exim's runtime configuration, you must edit **/usr/local/etc/exim/configure** and then "HUP" the Exim daemon (as root) - that is, send it a HUP (HangUP) signal. The daemon stores its process id (pid) in a file, in order to make this easy. Using this signal is less disruptive than completely stopping and starting the daemon. You can use these commands to send the signal (you must be root to do this):

```
# cat /var/run/exim.pid
# kill -HUP nnnn
```

where *nnnn* is the pid from the previous line. Alternatively, you can use the startup script installed by the port to do this for you:

```
# /usr/local/etc/rc.d/exim.sh reload
```

You can confirm that the daemon has restarted by checking the main log. Note that Exim installed it's startup script in /usr/local/etc/rc.d and not in /etc/rc.d - this is because Exim is a third party package and not part of the FreeBSD operating system.

The following sections contain some suggestions for configuration modifications that you can try, just to get a feel for how the configuration file works. You do not have to stick rigidly to these examples; use different domain names or user names if you want to.

Adding more local domains

Edit the configuration (/usr/local/etc/exim/configure), and change the **local_domains** setting. Remember in vi you can just do "/domainlist" to find the first and proceeding occurrences [using 'n'] of "domainlist" in the file. You want the setting to now look like this:

```
domainlist local_domains = @ : pcnn.pacnog.school.fj
```

where *nn* is the number of your host. Remember to HUP the daemon afterwards. Now you have a new local domain. Try sending it some mail:

```
$ mail yourname@pcnn.pacnog.school.fj
```

Check that it arrives in your mailbox.

If you want to add a lot of domains, or if you want to keep changing them, it is easier to keep the list of domains in a file instead of in the Exim configuration. (You can also keep them in several different kinds of database, such as LDAP or MySQL, but we don’t cover that in this workshop.)

4. Relaying from another host

In section 2 above, there is test to demonstrate that relaying is blocked if you connect to your host’s IP address.

We are now going to remove this block by changing a line in the configuration to let all the classroom hosts relay through your host. Change this line:

```
hostlist   relay_from_hosts = localhosts
```

to

```
hostlist   relay_from_hosts = 127.0.0.1 : nnn.nnn.nnn.0/27
```

where nnn.nnn.nnn.0/27 is the classroom network. If you don’t know what this is ask your instructor to remind the class. (Don’t forget to HUP the daemon.) Then try the telnet test from section 2 again. This time it should accept the request to relay. Ask one of the other students to try relaying through your host -- it should work. If you can, telnet from a host outside the classroom network, and confirm that relaying is still blocked.

Allowing relaying to specific domains

The default configuration contains the line

```
domainlist relay_to_domains =
```

This defines domains to which your host will relay, wherever the message comes from. As you can see, the default list is empty, so no domains match.

Add some domains to this line. For example, add the domain of your home email where I’ve put “somewhere.com”

```
domainlist relay_to_domains = somewhere.com
```

Now we need to test that Exim will indeed relay to those domains (but not to others) from a host that does not match **relay_from_hosts**. Exim has a testing facility that lets you simulate an SMTP call from a remote host. Run it like this:

```
$ exim -bh 192.168.1.1
```

You will see some debugging output, and then an SMTP greeting line. Now type SMTP commands, waiting for a response between each one:

```
ehlo testhost
mail from:<localuser@pcnn.pacnog.school.fj>
rcpt to:<user@somewhere.com>
rcpt to:<user@some.other.domain>
```

You will see the tests that Exim is making as it runs the ACL after each RCPT command. Check that it allows relaying to the right domains, and not to any others. End the SMTP session with the command "quit".

Last modified: Tue Jun 14 03:29:11 CLT 2005