

20

ANSWERS TO EVEN-NUMBERED EXERCISES

2. How would Max store a copy of his email in `~/mbox` and send a copy to `max@example.com`?

Max needs to create a `~/.forward` file with the following lines:

```
$ cat ~/.forward
~/mbox
max@example.com
\max
```

4. Describe how setting the `dc_eximconfig_configtype` variable in `/etc/exim4/update-exim4.conf.conf` to `smarthost` affects `exim4` behavior. What happens when you set this variable to `internet`?

Setting `dc_eximconfig_configtype` to `smarthost` causes `exim4` to

- Accept email originating on local systems for delivery to local systems.
- Accept email originating on local systems for delivery to nonlocal systems, delivering it using a smarthost to relay email.
- Not deliver email originating on nonlocal systems.
- Not relay email.

Setting `dc_eximconfig_configtype` to `internet` causes `exim4` to

- Accept email from local and nonlocal systems.
- Deliver email that originates on local systems to a local system or directly to a nonlocal system, without using a relay.
- Deliver email that originates on nonlocal systems to a local system only.
- Not relay email.

6. Describe the software and protocols used when Max sends an email to Sam on a remote Linux system.

One possible answer of many:

- a. Max's MUA passes the email to **exim4**.
 - b. The **exim4** daemon uses SMTP to deliver the email to the instance of **exim4** on Sam's system.
 - c. On Sam's system, **exim4** passes the email to the local MDA (**procmail**).
 - d. The MDA stores the email in Sam's file in the mail spool directory.
 - e. Sam's MUA retrieves the email from the mail spool directory.
8. Assume a script stores certain information in a variable named **RESULT**. What line could you put in the script that would send the contents of **RESULT** to the email address specified by the first argument on the command line?

```
echo "$RESULT" | exim4 $1
```

10. Describe the relationship between **spamassassin**, **spamd**, and **spamc**. How does each work? Why not use the **spamassassin** utility by itself?

The **spamassassin** utility is a stand-alone mail filter that rates and tags email for the likelihood of being spam. The **spamc** utility is a mail filter that is similar to **spamassassin** but more efficient and quicker to load.

The **spamc** utility is lighter weight than **spamassassin** because it offloads the work of evaluating each piece of email to the **spamd** daemon, which is always running in the background.

Loading a large program such as **spamassassin** for each piece of email a mail server receives imposes a substantial overhead on the server system. Because **spamc** calls **spamd**, which does the heavy lifting and is already loaded, **spamc** can be lightweight and impose a minimal overhead when it is called.