POP3/IMAP

UNIX ADMINISTRATION
WORKSHOP

Objectives

1. Definitions of IMAP, POP3, Exim, Courier-authlib, etc
2. Reconfigure Exim for Maildir delivery
3. Install Courier-authlib
4. Configure and start courier-authlib
5. Test courier-authlib

Objectives …

1. Install courier-imap
2. Configure and start courier-imap Test POP3 and IMAP
3. POP3 and IMAP over SSL

IMAP

IMAP is an **Internet Message Access Protocol**. It is a method of accessing electronic mail messages that are kept on a possibly shared mail server. In other words, it permits a "client" email program to access remote message stores as if they were local. For example, email stored on an IMAP server can be manipulated from a desktop computer at home, a workstation at the office, and a notebook computer while travelling, without the need to transfer messages or files back and forth between these computers. IMAP uses TCP/IP port 143.

POP

- Short for ***P****ost* ***O****ffice* ***P****rotocol,* a protocol used to retrieve e-mail from a mail server. Most e-mail applications (sometimes called an *e-mail client*) use the POP protocol, although some can use the newer IMAP (Internet Message Access Protocol).
- There are two versions of POP. The first, called *POP2,* became a standard in the mid-80's and requires SMTP to send messages. The newer version, POP3, can be used with or without SMTP. POP3 uses TCP/IP port 110.

POP3 vs IMAP

- With IMAP, all your mail stays on the server in multiple folders, some of which you have created. This enables you to connect to any computer and see all your mail and mail folders. In general, IMAP is great if you have a dedicated connection to the Internet or you like to check your mail from various locations.
- With POP3 you only have one folder, the Inbox folder. When you open your mailbox, new mail is moved from the host server and saved on your computer. If you want to be able to see your old mail messages, you have to go back to the computer where you last opened your mail.
- With POP3 "leave mail on server" only your email messages are on the server, but with IMAP your email folders are also on the server.

Exim

- Exim is an **open source** mail transfer agent (**MTA**), which is a program responsible for receiving, routing, and delivering e-mail messages (this type of program is sometimes referred to as an *Internet mailer*, or a *mail server program*). MTAs receive e-mail messages and recipient addresses from local users and remote **hosts**, perform **alias** creation and forwarding functions, and deliver the messages to their destinations. Exim was developed at the University of Cambridge for the use of **Unix** systems connected over the Internet. The software can be installed in place of **sendmail**, the most common MTA for UNIX and **Linux** systems. In comparison to sendmail, Exim is said to feature more straightforward configuration and task management.

Courier-authlib

- Courier is a mail system which includes a number of packages. It has its own MTA. We are interested in only the following components – IMAP/POP3 servers and sqwebmail
- The courier packages now share a single authentication library, courier-authlib. This package is responsible for looking up usernames and passwords

Secure Sockets Layer (SSL)

Secure Sockets Layer - The leading security protocol on the Internet. Developed by Netscape, SSL is widely used to do two things:

1. to validate the identity of a Web site
2. and to create an encrypted connection for sending credit card and other personal data.

- Look for a lock icon at the bottom of your browser when you order merchandise on the Web. If the lock is closed, you are on a secure SSL.

Maildir

- **Maildir** is a widely-used format for storing e-mail that does not require application-level file locking to maintain message integrity as messages are added, moved and deleted. Each message is kept in a separate file with a unique name. All changes are made using atomic filesystem operations so that the filesystem handles file locking concurrency issues. A Maildir is a directory (often named Maildir) with three subdirectories named tmp, new, and cur.

Reconfigure Exim for mail delivery

- We will configure Exim to deliver all local in Maildir format.
- /home/Maildir/new/cur/tmp
- Messages are written into tmp, moved to new when delivery is complete and moved to cur when read. Messages have a unique filename based on the hostname and time of day

Reconfigure Exim for mail delivery …

- Edit /usr/local/etc/exim/configure
- Find the local_delivery transport and modify it as follows:
- local_delivery:
- driver = appendfile
- **directory = $home/Maildir**
- **maildir_format**
- **maildir_use_size_file**
- **#file = /var/mail/$local_part**
- delivery_date_add
- envelope_to_add
- return_path_add
- group = mail
- **#user = $local_part**
- mode = 0660
- no_mode_fail_narrower

Reconfigure Exim for mail delivery ..

- Optionally you could add further parameters to this transport which let you impose quotas on your users, for example to limit all users to 10 megabytes of storage each:
- **maildir_tag = ,S=$message_size**
- **quota_size_regex = ,S=(\d+)**
- **quota = 10M**
- **quota_warn_threshold = 90%**

Install Courier-authlib

- # **cd /usr/ports/security/courier-authlib/**
- # **make**
- When prompted for options on the screen, press the down arrow to highlight the option:
- **[X] AUTH_USERDB Userdb support**
- Press <TAB> to highlight OK, and then <ENTER> to continue.
- # **make install**
- # **make clean** (optional step - deletes temporary files in 'work' subdir)
- Total compile time on your machines will be between 10 and 15 minutes.

Configure and start courier-authlib

- Remember to HUP your exim daemon. Now test out your new configuration by delivering to some local account on your machine:
- $ **/usr/local/sbin/exim -bt *localuser***
- localuser@PCN.ws.linuxchix.or.ke
- router = localuser, transport = local_delivery
- $ **/usr/local/sbin/exim *localuser***

Configure and start courier-authlib

- **Here is a test**
- **.**
- $ **cd /home/*localuser*/Maildir**
- $ **ls**
- cur new tmp
- $ **ls new**
- 102078119.7969. PCN.ws.linuxchix.or.ke,S=426
- $ **cat new/***
- Return-path: <root@ PCN.ws.linuxchix.or.ke >
- ...
- Here is a test

Test courier-authlib

- courier-authlib runs a pool of authentication daemons which perform the actual work; courier-imap and sqwebmail communicate with these daemons via a socket. So the next thing we need to do is to start the daemons. First you need to edit /etc/rc.conf:
- # **vi /etc/rc.conf**
- add the following line:
- **courier_authdaemond_enable="YES"**
- Courier-authlib itself has a single configuration file, /usr/local/etc/authlib/authdaemonrc. For the purposes of this exercise, we will turn on authentication debugging.
- # **cd /usr/local/etc/authlib**
- # **vi authdaemonrc**
- change this line:
- DEBUG_LOGIN=0
- to:
- **DEBUG_LOGIN=1**

Test courier-authlib …

- To save resources, you can also configure the authdaemond process not to try any authentication mechanisms which you know you don't need. For example, if all your authentication is only via PAM for Unix system passwords, then you can remove all the others. Save the original line so that your changes look like this:
- **#authmodulelist="authuserdb authvckpw authpam authldap authmysql authpgsql"**
- **authmodulelist="authpam"**
- **Now we are ready to start the authentication daemons:**
- **# /usr/local/etc/rc.d/courier-authdaemond start**

- **Starting courier_authdaemond.**
- **# ps auxwww | grep authdaemond**
- **ps shows one courierlogger process, and six authdaemond processes (one master, five workers). If you didn't see "Starting courier_authdaemond" then you made a typing error.**

Test courier-authlib …

- You can test the authentication system by itself; the "authtest" command sends requests down the authentication socket, and displays the responses which come back. Test using any Unix login account which already exists on your system.
- # **authtest brian** -- find an account called 'brian'
- # **authtest brian foo** -- check 'brian' has password 'foo'
- # **authenumerate** -- list all accounts
- Try it also with a non-existent username, and with both the right password and a wrong password for an account, to confirm that passwords are being validated properly.
- Because we enabled login debugging, you should find that each authentication request generates detailled information in /var/log/debug.log showing how the request is passed to each module in turn. Have a look in this file to confirm:
- # **less /var/log/debug.log**

Install courier-imap

- Using ports, building courier-imap is straightforward:
- **# cd /usr/ports/mail/courier-imap**
- **# make**
- [When prompted for options on the screen, press <TAB> to highlight OK, and then <ENTER> to continue.]
- # make install
- # make clean (optional step)
- Compilation will take 10 to 15 minutes on your machines.

Configure and start courier-imap

- You can choose to run POP3, IMAP, or both. There is a configuration file for each one:
- /usr/local/etc/courier-imap/pop3d
- /usr/local/etc/courier-imap/imapd
- The default configuration is acceptable in most cases. However for a large server you may wish to increase the maximum number of concurrent connections from the default of 40, if you have fairly powerful hardware:
- # **cd /usr/local/etc/courier-imap**
- # **vi pop3d**
- ...
- MAXDAEMONS=300
- ...
- # **vi imapd**
- ...
- MAXDAEMONS=300

Configure and start courier-imap

- Then, you need to enable the daemon(s) which you wish to run in /etc/rc.conf
- **# vi /etc/rc.conf**
- add the following line(s):
- courier_imap_pop3d_enable="YES"
- courier_imap_imapd_enable="YES"
- And then run the startup script(s):
- **# /usr/local/etc/rc.d/courier-imap-pop3d.sh start**
- Starting courier_imap_pop3d.
- **# /usr/local/etc/rc.d/courier-imap-imapd.sh start**
- Starting courier_imap_imapd.

Test POP3 and IMAP

- Test using telnet: POP3 and IMAP are both text-based layer 7 protocols and you can drive them by hand.
- **# telnet localhost 110**
- Connected to localhost.ws.afnog.org
- Escape character is '^]'.
- +OK Hello there.
- user *username*
- +OK Password required.
- pass *password*
- +OK logged in.
- stat
- +OK 26 49857
- retr 1
- +OK 1073 octets follow.
- ... message
- .
- quit
- +OK Bye-bye.
- Connection closed by foreign host.

Test POP3 and IMAP …..

- # **telnet localhost 143**
- Connected to localhost.ws.afnog.org.
- Escape character is '^]'.
- * OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT
- THREAD=REFERENCES SORT QUOTA IDLE ACL ACL2=UNION STARTTLS] Courier-IMAP ready.
- Copyright 1998-2005 Double Precision, Inc. See COPYING for distribution information.
- **a login** *username password*
- a OK LOGIN Ok.
- **a examine inbox**
- * FLAGS (\Answered \Flagged \Deleted \Seen \Recent)
- * OK [PERMANENTFLAGS ()] No permanent flags permitted
- * 26 EXISTS
- * 0 RECENT
- * OK [UIDVALIDITY 989061119] Ok
- * OK [READ-ONLY] Ok
- **a logout**
- * BYE Courier-IMAP server shutting down
- a OK LOGOUT completed
- Connection closed by foreign host.

Test POP3 and IMAP …..

- **NOTE:** The daemons will fail to login if the mail directory does not exist, although current versions do now provide an error message. Hence you need to have delivered at least one message to the user, to create their mailbox, before they can login (or use the 'maildirmake' command to create it). Look for logging messages in /var/log/maillog and /var/log/debug.log.

POP3 and IMAP over SSL

- If you wish, you can choose to allow pop3 over SSL (port 995) and imap over SSL (port 993). The advantage is that, for clients which support it, the traffic is encrypted. The disadvantage is higher CPU load on your server for the encryption of data.
- To run SSL you will need a certificate. For testing purposes you can use a 'self-signed' certificate. The pop3d.cnf and imapd.cnf files contain the
  parameters for the Snakeoil certificate. You may edit this for you environment, but note that it is not proper certificate signed by a recognised CA.
  Run the following scripts which will generate them for you:
- # **cd /usr/local/etc/courier-imap**
- **# cp pop3d.cnf.dist pop3d.cnf**
- **# cp imapd.cnf.dist imapd.cnf**
- # **mkpop3dcert**
- # **mkimapdcert**
- Next, enable the SSL daemons in /etc/rc.conf:
- # **vi /etc/rc.conf**
- **courier_imap_pop3d_ssl_enable="YES"** # pop3 over ssl, port 995
- **courier_imap_imapd_ssl_enable="YES"** # imap over ssl, port 993

POP3 and IMAP over SSL

- Then you start the servers:
- # **/usr/local/etc/rc.d/courier-imap-pop3d-ssl.sh start**
- Starting courier_imap_pop3d_ssl.
- # **/usr/local/etc/rc.d/courier-imap-imapd-ssl.sh start**
- Starting courier_imap_imapd_ssl.
- You can't use a regular telnet to test it, because all your communication needs to be encrypted, but openssl has an SSL client you can use to make an encrypted connection for testing:
- # **openssl s_client -connect localhost:993**
- # **openssl s_client -connect localhost:995**
- See the previous exercise for the commands to use with each connection.
- If you were running the service commercially you might want to consider a certificate signed by a recognised CA, rather than using a self-signed certificate.