Multiple Personalities

IMAP

MIKE BRODBELT

I've been using electronic mail for some years now, and, like many other Linux users, I have more than one email address. Currently, I have about six accounts that I actively use, and several dozen other addresses that deliver to me. I used to use my email to manage my work — these days managing my email has become my work, along with managing mail for my network.

For anyone who has a requirement to access multiple separate mail accounts that reside on different machines, access to email becomes problematic. Many mail systems use the tried and tested POP3 protocol. POP3 client applications download the mail from the server, and store it on the client computer. Most clients store mail in their own format, making it inaccessible to other mail programs, and most machines which run POP3 clients are desktops, which are rarely on 24/7, further reducing access to the mail once it has been downloaded.

The IMAP protocol attempts to remedy some of these problems. The strength of IMAP (Internet Message Access Protocol) lies in online and disconnected operation. Unlike POP3, mail is not copied from the server and then deleted. Instead, IMAP clients manipulate the mail on the server, and permit access to remote, server hosted mailboxes as though they were local resources.

An IMAP mail system has a number of immediate advantages for users:

As all mail is stored on the server, changing mail client becomes the work of seconds. All that is required is to configure a new IMAP client with the IMAP account details.

An IMAP client can easily be configured to view multiple mailboxes in physically separate servers.

Multiple IMAP clients can be used by each user. This makes implementing a Web mail solution for roaming users a simple task.

IMAP maintains message status flags on the server for read, answered, etc.

IMAP allows shared folders. This makes it easier to implement generic email accounts for an organisation, and then allow multiple users to access those accounts.

Many implementations also allow server side filtering of mail. This can be an extremely useful feature when users are accessing their mailboxes through different email clients.

As of this writing, the IMAP capabilities defined are:

ACL	*[RFC2086]*
IDLE	*[RFC2177]*
LITERAL+	*[RFC2088]*
LOGIN-REFERRALS	*[RFC2221]*
MAILBOX-REFERRALS	*[RFC2193]*
NAMESPACE	*[RFC2342]*
QUOTA	*[RFC2087]*
UIDPLUS	*[RFC2359]*
STARTTLS	*[RFC2595]*
LOGINDISABLED	*[RFC2595]*
ID	*[RFC2971]*

Software

There are a number of IMAP servers available, but this discussion is limited to those which fall under some sort of open-source license. IMAP offers a number of extensions to the basic protocol, and different servers implement different subsets of this functionality.

Detailed information about any of these can be gleaned from the appropriate RFC, but those likely to be of most interest are ACL (access control list) support, which offers fine grained control over user access to mailboxes, QUOTA support, which permits mailbox level quotas independent of any disk quota scheme in use, and STARTTLS, which allows IMAP over SSL secured connections.

The three main open-source IMAP servers in use are:

Courier IMAP (*http://www.inter7.com/courierimap/*), University of Washington (UW) IMAP

(*http://www.washington.edu/imap/*), and Cyrus (*http://asg.web.cmu.edu/cyrus/imapd/*). These servers all offer slightly different feature sets, and which is best will depend entirely on the demands of the user base it is expected to serve.

The choice of IMAP server may well be dictated by the MTA (Mail Transfer Agent) in use, as both the MTA and the IMAP server must understand a common mailbox format. The UW server offers no support for maildir, and has no plans to do so at the time of writing. The Courier server was specifically written to allow IMAP access to maildir format mailboxes, and so users of Qmail (which uses maildir) will find Courier to be their only choice of IMAP server at this time. The Cyrus server supports only its own format, but provides with the distribution a local delivery agent that can understand this format, so integrating it into most MTA's should be possible. The UW server supports several mailbox formats, so if access to mail via Elm or any other mail client that reads the mailbox directly is required, then UW will be the server of choice.

If the choice of server has not already been made by the above paragraph, other features may be important. The Cyrus server allows you to run it as a black-box system. Cyrus users need neither shell access to the IMAP server, nor an account in /etc/passwd. Users of UW or Courier servers need accounts in /etc/passwd to receive mail. Cyrus implements the IMAP ACL and QUOTA extensions, UW relies on OS level disk quotas, and thus can generate hard bounces for over-quota situations.

For my purposes, I chose to use the Cyrus server. Cyrus is a feature rich server, and supports several features, which I consider important. It implements the IMAP ACL and QUOTA extensions, which give it great administrative flexibility. It has full support for several encrypted authentication methods, via the Cyrus-SASL library (see below), and it supports IMAP over SSL.

Compiling and installing the Cyrus server

There are binary distributions of Cyrus available in rpm or deb format, and installing one of these may well represent the simplest way to get the Cyrus server installed. Nevertheless, I chose to compile the server from source as this provides far more flexibility that a precompiled distribution, and, with a package as complex and powerful as Cyrus, the time invested in customising the setup for your needs is time well spent.

While Cyrus is an excellent package, the documentation left much to be desired. The first problem to be faced in attempting to compile Cyrus was the Cyrus SASL (Simple Authentication and Security Layer) library.

Recent versions of Cyrus (version 2 or greater) require the Cyrus SASL authentication library to be installed before the IMAP server. SASL is an authentication multiplexer — it can be compiled to use a number of authentication methods, and it hides the details of these authentication methods from the application using them. A site may have a number of applications that use SASL, and these applications need only be written to authenticate via SASL. The SASL library can be built to authenticate via Kerberos, GSSAPI, CRAM-MD5, DIGEST-MD5, and others. SASL provides the option of storing authentication information within a Berkeley database on disk, for those who do not have a Kerberos or similar infrastructure in place. If this is to be used, it is important that the SASL library and the applications using SASL be compiled with the same version of libdb. SASL will happily compile with the version included with glibc on most systems, but Cyrus IMAP will not, and requires Berkeley DB.

The Berkeley DB package can be downloaded from *http://www.sleepycat.com*. I installed it in */usr/local/BerkeleyDB.3.2/* and then configured SASL to use it:

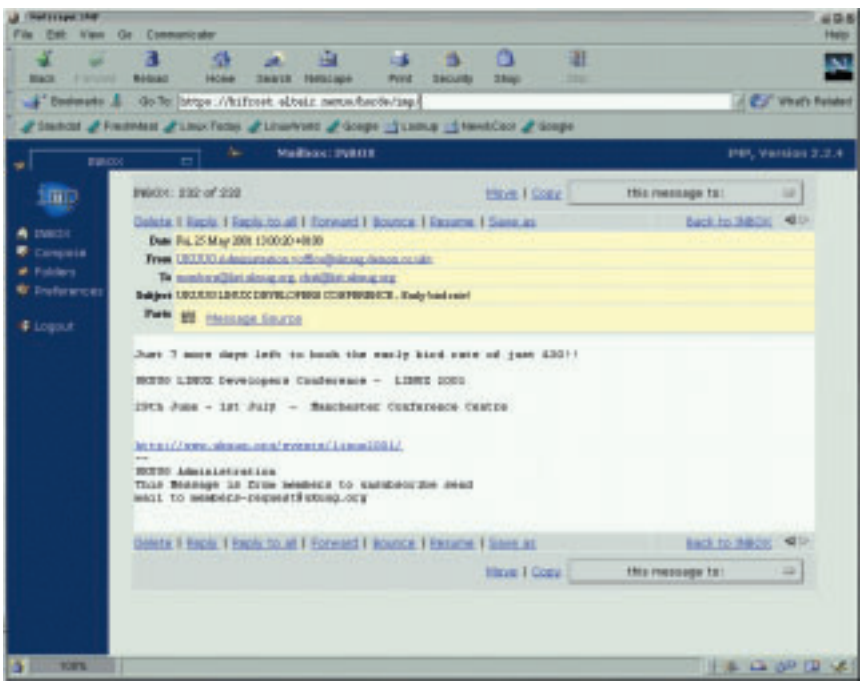
```
# export LIBRARY_PATH=/usr/local/BerkeleyDB➥
.3.2/lib/
# export C_INCLUDE_PATH=/usr/local/Berkeley➥
DB.3.2/include/
# export LDFLAGS=-R/usr/local/BerkeleyDB➥
.3.2/lib/
# ./configure --prefix=/usr --disable-gssapi➥
 --disable-krb-4➥
  --with-pam=yes --with-dblib=berkeley --with-➥
rc4=/usr/local/ssl/
```

This configuration was for a test system with no Kerberos or GSSAPI authentication, with OpenSSL 0.9.6. OpenSSL should be compiled to generate a shared library. This compiles a SASL library with support for anonymous, CRAM-MD5, DIGEST-MD5, and PLAIN authentication methods. Any application compiled against the SASL library will now be able to

BOX 1

configdirectory: /var/imap
partition-default: /var/spool/imap
admins: cyrus
sendmail: /usr/sbin/sendmail

IMAP client IMP



Mutt as the client

offer any of these authentication methods to client applications. This can provide significant additional security — PINE and Mutt both support CRAM-MD5 authentication, which obviated the need to send authentication credentials in the clear. The SASL architecture allows more authentication methods to be plugged-in to SASL as they are developed.

Once the SASL library is installed, Cyrus can be compiled relatively easily, though it is important to remember to add a user account for Cyrus to run under to /etc/passwd before compiling the server. The version I used was 2.0.11, and it was configured as follows:

```
# ./configure --prefix=/usr --sysconfdir=/etc ➥
--localstatedir=/var ➥
--with-openssl=/usr/local/ssl/ --with-sasl=/➥
usr/lib/sasl/ ➥
--without-krb --with-dbdir=/usr/local/Berkeley➥
DB.3.2/
```

To make Cyrus compile correctly, I had to make two small alterations. I added a symbolic link from */share* to */usr/share* (without this the compile_et program caused the compile to fail), and I also had to copy the ssl shared libraries from */usr/local/ssl/lib* to */usr/lib* before the compilation found them.

Configuring and testing the Cyrus server

After installation, there are several steps necessary to get you new IMAP server up and running. First, create your */etc/imapd.conf* file. This is a simple configuration file, and a basic setup should look something like the file in Box 1. For a full description of the fields in this file, see the imapd.conf(5) man page.

Next, create the "configdirectory" specified in the imapd.conf file. Ensure this is owned by the Cyrus user and group (by default, cyrus:mail), and change its permissions to 750. Do the same for the "partition-default" directory. Then, run the tools/mkimap script from the Cyrus distribution, as the Cyrus user - this will create the Cyrus directories under those you just created. On Linux file systems (ext2 - this does not apply to ReiserFS, XFS, or similar), it's important to use the "chattr +S" command to set these directories and their contents for synchronous updates. The ext2 filesystem can be prone to mailbox corruption under certain circumstances without this attribute set. Using synchronous updates forces the operating system to flush changes to these directories to the disk immediately, and generates a performance overhead. For a large system, it may be preferable to use a journaling filesystem to obviate the need for this.

Ensure that your /etc/services file contains all the entries in Box 2.

BOX 2

pop3	*110/tcp*
imap	*143/tcp*
imsp	*406/tcp*
acap	*674/tcp*
imaps	*993/tcp*
pop3s	*995/tcp*
kpop	*1109/tcp*
sieve	*2000/tcp*
lmtp	*2003/tcp*
fud	*4201/udp*

Finally, the master process must be configured. The Cyrus distribution comes with a number of sample configurations in the *master/conf* directory. Choose the appropriate one, and copy it to */etc/cyrus.conf*, and uncomment the entries required.

To test connections to the IMAP server, start the master process and try to telnet to the server on the IMAP port:

```
$ telnet bifrost 143
Trying 192.168.1.4...
Connected to bifrost.altair.nexus.
Escape character is '^]'.
* OK bifrost.altair.nexus Cyrus IMAP4➥
 v2.0.11 server ready
```

If you see a greeting message like that above, your server is running. Add a user and password to your SASL secrets file using the saslpasswd utility (this won't be necessary if you already have an authentication framework like Kerberos in place). You can then test connections for this user with the imtest script from the Cyrus distribution:

```
# /usr/bin/imtest -m login -a imapuser bifrost
C: C01 CAPABILITY
S: * OK bifrost.altair.nexus Cyrus IMAP4➥
 v2.0.11 server ready
S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA➥
 LITERAL+ NAMESPACE UIDPLUS ID➥
 NO_ATOMIC_RENAME UNSELECT MULTIAPPEND SORT➥
 THREAD=ORDER
EDSUBJECT THREAD=REFERENCES IDLE AUTH=DIGEST➥
-MD5 AUTH=CRAM-MD5
S: C01 OK Completed
Password:
C: L01 LOGIN imapuser {9}
+ go ahead
C: <omitted>
L01 OK User logged in
Authenticated.
Security strength factor: 0
. logout
* BYE LOGOUT received
. OK Completed
Connection closed.
```

At this stage, you have a working IMAP server installed. You now need to add user mailboxes. This is done with a Perl program called Cyradm, which is installed as part of the Cyrus distribution. This should be run as the Cyrus user, and allows a number of administrative operations:

```
$ cyradm bifrost.altair.nexus
Please enter your password:
bifrost.altair.nexus> ?
authenticate, login, auth      authenticate➥
 to server
chdir, cd                      change current➥
 directory
createmailbox, cm, create      create mailbox
deleteaclmailbox, dam,➥
 deleteacl                     remove ACLs➥
 from mailbox
deletemailbox, delete, dm      delete mailbox
disconnect, disc               disconnect➥
 from current server
exit, quit                     exit cyradm
```

```
help, ?                          show commands
listacl, lam, listaclmailbox     list ACLs on
 mailbox
listmailbox, lm                  list mailboxes
listquota, lq                    list quotas
 on specified root
listquotaroot, lqr, lqm          show quota
 roots and quotas for mailbox
renamemailbox, rename, renm      rename (and
 optionally relocate) mailbox
server, servername, connect      show current
 server or connect to server
setaclmailbox, setacl, sam       set ACLs on
 mailbox
setquota, sq                     set quota on
 mailbox or resource
version, ver, info               display
 version info of current server
```

Each user should have a mailbox created. For the imapuser test user, create a mailbox called user.imapuser through Cyradm. This will become the INBOX for that user. All other mailboxes will be subordinate to this one, and are best created via a mail client.

To complete the installation, you need to arrange for the Cyrus master process to start when the system boots, and also configure your MTA to deliver mail into the Cyrus mailstore. Cyrus provides a local delivery agent and the MTA must be configured to call for local mail. The Cyrus documentation provides information on how to achieve this with sendmail. For other MTA's, different procedures will be required.

Conclusion

It should be clear from the above that a Cyrus installation provides a powerful and flexible mail system. Installing and configuring it correctly is not a simple process, but once set up, it provides a vastly superior alternative to the traditional POP3 mailbox setup. Most mail clients can be configured to use multiple IMAP accounts, so for users with many mailboxes IMAP simplifies mail handling immensely. All your mail accounts can be handled from anywhere with an IMAP client. Several IMAP Web mail clients exist, these can vastly simplify life for roaming users, who can then access their mail from anywhere with an Internet connection. Examples of these are Squirrel Mail (*http://www.squirrelmail.org*) and IMP (*http://www.horde.org/imp/2.2*).

Both of these require additional configuration work, as they are PHP based. Squirrel Mail uses its own implementation of the IMAP protocol, and, for this reason is probably easier to set up than IMP, which requires an external library distributed with the UW IMAP server. The screenshots show three IMAP clients in use — Netscape, Mutt, and IMP. All three are using different forms of encrypted authentication, and are viewing the same mailbox.

BOX 3

IMAP over SSL

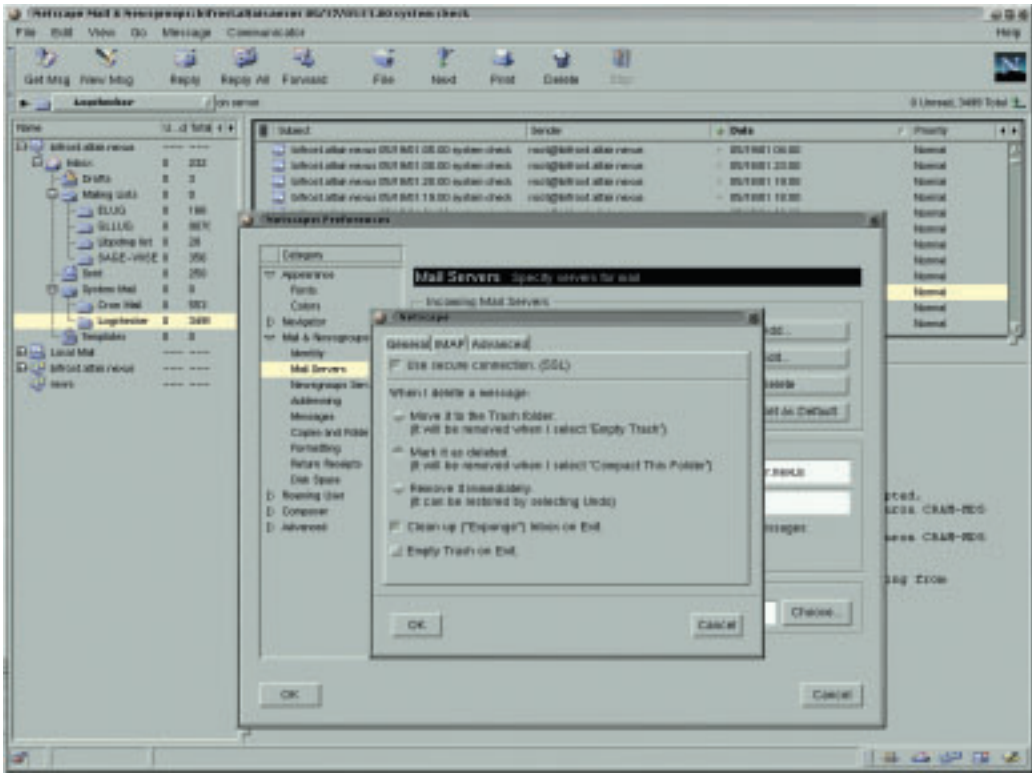
Enabling SSL support in Cyrus can be simply achieved by creating a self signed X509 certificate and private key pair with OpenSSL:

```
$ openssl req -new -x509 -nodes -out /var/
        imap/server.pem
 -keyout /var/imap/server.pem -days 365
```

Then, uncomment the imaps service definition in */etc/cyrus.conf*, and add the following lines to */etc/imapd.conf*:

```
tls_cert_file: /var/imap/server.pem
tls_key_file: /var/imap/server.pem
```

After restarting the imap server, SSL support will be enabled. ■



Mail server settings for Netscape