The Exchange Server Replacement HOWTO

Curt Johnson
CEO
Array Services, Inc.

Edited by

Chuck Mead

The Exchange Server Replacement HOWTO

by and Curt Johnson
Edited by Chuck Mead

Abstract

This document describes the installation and configuration of an IMAP and POP3 mail server using LDAP as the user database. Hopefully it will answer more questions than it creates. Improvements, constructive criticism, additions and corrections are gratefully accepted. Please mail your feedback to the author, with "Exchange Replacement HOWTO" in the subject.

Copyright and Disclaimer

Caution

No liability for the contents of this document can be accepted. Use the concepts, examples and other content at your own risk. Additionally, this is an early version, possibly with many inaccuracies or errors.

A number of the examples and descriptions use the Red Hat package layout and system setup. Your mileage may vary. As far as I know, only programs that, under certain terms may be used or evaluated for personal purposes will be described. Most of the programs will be available, complete with source, under the terms of the GNU (http://www.gnu.org/copyleft/gpl.html) Public License.

Revision History
Revision 0.516 November 1999

Revision 0.619 December 1999

Table of Contents

List of Examples

Chapter 1. Introduction

This document is meant to document the replacement of a Microsoft Exchange server with a Linux based solution. Currently this information is for Red Hat 6.1 systems as this seems to be the only system with a packaged, working version of the pam_ldap module. If you have a work around for other systems please contact me. The functionality you should expect when configured includes, LDAP address book, POP3 and IMAP4 accounts and mail user management through LDAP. Hopefully this document will expand to cover more features and some more system administration utilities.

1.1. New Versions of this Document

The permanent home for this document can be found at
http://www.arrayservices.com/projects/Exchange-HOWTO/
(http://www.arrayservices.com/projects/Exchange-HOWTO/)

1.2. Feedback

All comments, error reports, additional information and criticism of all sorts should be directed to:

cjohnson@arrayservices.com (mailto:cjohnson@arrayservices.com)

Note: Please be sure and include “Exchange Server”, or "HOWTO" in your subject.

1.3. Further Reading

The Sendmail FAQ (http://www.sendmail.org/faq/)

LDAP HOWTO (http://www.linuxdoc.org/HOWTO/LDAP-HOWTO.html)

Cyrus IMAP mini-HOWTO (http://www.linuxdoc.org/HOWTO/mini/Cyrus-IMAP.html)

Mail HOWTO (http://www.linuxdoc.org/HOWTO/Mail-Administrator-HOWTO.html)

PAM FAQ (http://www.us.kernel.org/pub/linux/libs/pam/FAQ)

Chapter 2. Software

2.1. Cyrus IMAP Server

You will need two components for the Cyrus IMAP server, the server cyrus-imap and the SASL libraries for authentication. The Cyrus project's home page is http://asg.web.cmu.edu/cyrus/. From this page you can obtain the latest compressed tar files for the imap server and the SASL libraries. Alternatively and more convenient, RPMs are available. As of this writing, cyrus-imapd-1.6.1-2.i386.rpm and cyrus-sasl-1.5.3-2.i386.rpm are the latest Red Hat RPMs available from http://www.rpmfind.net. There are libc6 contributed rpms available at this location that are more recent. The latest rpm releases are cyrus-imapd-1.6.13-1.i386.rpm and cyrus-sasl-1.5.5-1.i386.rpm

2.2. Sendmail

The standard sendmail installed with most distributions should work just fine. Later in this document we will create a new sendmail.mc and generate a new sendmail.cf but the installation of sendmail is pretty straightforward. If you choose to download and compile sendmail you can choose the default configuration and install options. If you are installing from RPM please make sure you have both the sendmail and sendmail-cf packages installed. The current RPMs as of time of writing are sendmail-8.9.3-15.i386.rpm and sendmail-cf-8.9.3-15.i386.rpm.

2.3. OpenLDAP

The OpenLDAP project is located at http://www.openldap.org. Compressed archive files are available from there, or you may use RPMs supplied by Red Hat.

The current RPM is openldap-1.2.7-2.i386.rpm. The only major difference between the RPM and compiling the libraries yourself is the location of the configuration files. A default compile and install will place the configuration files in /usr/local/etc/openldap whereas the RPMs will place the configuration files in /etc/openldap. The RPM will also install a startup script in /etc/rc.d/init.d for you. If you choose to build the OpenLDAP package yourself, you will have to generate your own script. It is advantageous to use the RPM for this portion of the setup if you intend to run the GQ LDAP administration GUI on the server machine. GQ requires the openldap RPM to be installed on the machine that it will be run from.

2.4. GQ LDAP GUI

GQ is a gtk application that allows you to setup configurations for multiple servers and search, as well as browse the entries. GQ will also allow you to add entries by using existing entries as templates. By binding to the LDAP server with the bind dn, you can add, edit and delete entries from the GUI. GQ is limited in that is cannot, to my knowledge, add fields to an entry. GQ's home page is http://www.biot.com/gq/. Compressed archive files of the source are available from this location, an RPM file is available from ftp://ftp.ujep.cz/pub/OS/Linux/local/RPMS/i386/ (ftp://ftp.ujep.cz/pub/OS/Linux/local/RPMS/i386/). At the time of this writing, the current RPM file was gq-0.2.2-1.i386.rpm. GQ requires GTK 1.2+ and the openldap libraries. You may specify the location of the LDAP libraries if you are compiling, otherwise the RPM file will expect the libraries installed with the OpenLDAP RPM file. Please check the GQ home page for more information.

2.5. PAM LDAP module

This is probably the most difficult portion the the setup. The reason the this document currently only covers Red Hat 6.1 is the availability of a pam_ldap

RPM. This package is pam_ldap-36-1.i386.rpm. The home page for the current maintainer of the pam_ldap module is http://www.padl.com/pam_ldap.html. A compressed archive file of the source is available here, currently the file is pam_ldap.tgz and expands into a pam_ldap-42 directory.

In order to compile the pam_ldap module you will need the LDAP libraries. The documentation provided with the source recommends the Netscape LDAP C SDK, which is available from Netscape's developer website http://developer.netscape.com. I could not get pam_ldap to compile with this code. I also tried the Open LDAP development package, with which the pam_ldap Makefile has options for compatibility. The only package that seemed to work for me was a CVS download of the LDAP package from http://www.mozilla.org. Hopefully some light may yet be shined upon this module. If you compile the pam_ldap module yourself, you will need to create a configuration file, /etc/ldap.conf. This file is provided with the RPM file. I highly recommend use of the RPM unless you are brave at heart and have plenty of free time.

Chapter 3. Installation

3.1. RPM

RPM installation is pretty straightforward, download the RPMs to a directory, su to root, and type rpm -ivh *rpm. As long as your dependencies are installed, this will install the RPMs and you can start the configuration process.

3.2. Cyrus IMAP Server

If you have downloaded the Cyrus imap server and the SASL libraries, unpack them into some working directory. First you will need to compile the SASL libraries. Change directories to the top level of the SASL directory and run ./configure. I usually run ./configure with the –disable-krb4 –disable-gssapi since most Red Hat systems do not have Kerberos built in by default and I do not want to install it on my system. Next run make and make install. After you run make install, you may want to create a symbolic link from /usr/local/lib/sasl to /usr/lib/sasl by typing:

```
ln -s /usr/local/lib/sasl /usr/lib/sasl
```

The next step is to create a cyrus user. The simplest way for inexperienced administrators to do this is by using the linuxconf utility, simply type linuxconf –text at the command line while you are root. Using the tab key and arrows, you should be able to find the user management menu. Add a user with the name cyrus whose primary group is mail. This user should not have any administrative privileges. You will need to su to cyrus in order to finish the installation so do give the user a shell. You should probably not allow this user telnet access or any other remote access as well.

Next install the cyrus imap server, change directories to the top directory of the cyrus-imap package and run ./configure. I usually enable the

–enable-netscapehack option. Next run make depend, make and make install.

3.3. OpenLDAP

The OpenLDAP installation is pretty simple, download the package, unpack it and change directories to the top of the archive. Next run ./configure then make depend, make and make install. You will need to create a startup script in /etc/rc.d/init.d/ and symbolic links to the run levels that you want it to start on, we will do this during configuration.

3.4. LDAP libraries for pam_ldap

You're kind of on your own here. You can download and compile which ever packages work for you, the Netscape LDAP C SDK, the mozilla LDAP package from CVS, or the Open LDAP development libraries. I have attempted this with Netscape LDAP C SDK, archives of the mozilla LDAP module and the Open LDAP libraries. The only one that worked for me was pulling the mozilla LDAP module from CVS. Please check out the mozilla LDAP module page on the mozilla website (http://www.mozilla.org) and the CVS man page for instructions on how to do this. Anyone who would like to clean up this process, please let me know what you find out.

3.5. PAM LDAP module

After you download and unpack the pam_ldap archive, change directories to the top of the unpacked archive. You will need to edit the Make.defs file to match the libraries you compiled. The following lines will need to be edited. The CDEFS line appears in several places and can be commented out and uncommented depending on the libraries you are compiling with. The following lines are for the mozilla LDAP module, which I compiled and installed the libraries libldap.so and liblber.so in /lib/. The CVS code I downloaded into my user directory.

Example 3-1. The Mozilla LDAP Module

```
LDAP_LIB_DIR= /lib

LDAP_INC_DIR= <732>/ldap-cvs/mozilla/dist/Linux-2.2.10_x86_DBG.OBJ/include

LDAPLIBS= -L$(LDAP_LIB_DIR) -llber -lldap

CDEFS= -g $(WARNINGS) -D_REENTRANT -DLDAP_VERSION3_API -DNETSCAPE_API_EXTENSIONS
```

After you have configured your Make.defs file simply run make -f Makefile.linux and make -f Makefile.linux install. This will install the pam_ldap.so into /lib/security and modify the permissions. The only step left, will be to edit the proper files in /etc/pam.d/ so that the imap and pop services use the pam_ldap module to authenticate users.

3.6. Sendmail

After downloading and unpacking the sendmail archive, change directories to the top level directory of the unpacked archive. Follow the directions in the FAQ at http://www.sendmail.org/faq/ and build a default installation of Sendmail. If you run into trouble, simply use the RPMs for sendmail, all security and most other features are handled in the configuration files for sendmail.

3.7. GQ LDAP GUI

Be nice to yourself, use the RPM. If you have compiled the Open LDAP software yourself you will need to use the –with-ldap-prefix=/path/to/ldap with the ./configure to compile gq. The RPM will complain about libraries such as /usr/lib/libldap.so.1 and /usr/lib/liblber.so.1 simply make symbolic links from /lib/libldap.so and /lib/liblber.so to satisfy the dependencies.

Chapter 4. Configuration

4.1. DNS

If this server will be your main mail server for both outgoing and incoming mail, then you will need to make sure that there is an MX record in your DNS server that points to it, or that your server is configured to the address pointed to by the MX record. For more information on configuring DNS, please see the DNS HOWTO (http://www.linuxdoc.org/HOWTO/DNS-HOWTO.html).

4.2. inetd.conf

First we will configure your /etc/inetd.conf file. If you installed using RPMs, this may already be done for you. The cyrus imap server is called by inetd rather than running as a separate daemon. There should already be two lines that start with pop-3 and imap in your file. Edit the lines to match the following:

Example 4-1. inetd.conf

```
pop-3 stream tcp nowait cyrus /usr/sbin/tcpd /usr/cyrus/bin/pop3d

imap stream tcp nowait cyrus /usr/sbin/tcpd /usr/cyrus/bin/imapd
```

The entry /usr/sbin/tcpd tells inetd to use Tcp Wrappers with these servers. You can configure the files /etc/hosts.allow and /etc/hosts.deny to control which ip addresses and host names are allowed. For more information read the man files for hosts.allow and hosts.deny.

4.3. Open LDAP configuration

4.3.1. ldap startup script

The ldap startup script is included in the Open LDAP RPM. If you compiled Open LDAP you will need to create your own and add the symbolic links to your run levels. The Open LDAP RPM startup script for Red Hat (which uses the Red Hat initscripts functions) follows:

Example 4-2. Open LDAP RPM Startup Script for Red Hat Linux

```
#!/bin/sh
#
# ldap  This shell script takes care of starting and stopping
#       ldap servers (slapd and slurpd).
#
# chkconfig: - 70 40
# description: LDAP stands for Lightweight Directory Access Protocol, used \
#              for implementing the industry standard directory services.
# processname: slapd
# config: /etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.

[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/sbin/slapd ] || exit 0
```

```
[ -f /usr/sbin/slurpd ] || exit 0
RETVAL=0
# See how we were called.
case "$1" in
    start)
        # Start daemons.
        echo -n "Starting ldap: "
        daemon slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                daemon slurpd
                RETVAL=$?
                [ $RETVAL -eq 0 ] && pidof slurpd | cut -f 1 -d " " > /var/run/slurpd
            fi
        fi
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
        ;;
    stop)
        # Stop daemons.
        echo -n "Shutting down ldap: "
        killproc slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                killproc slurpd
                RETVAL=$?
            fi
        fi
        echo
        if [ $RETVAL -eq 0 ]; then
            rm -f /var/lock/subsys/ldap
            rm -f /var/run/slapd.args
        fi
        ;;
    status)
```

```
        status slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                status slurpd
                RETVAL=$?
            fi
        fi
        ;;
    restart)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
    reload)
        killproc -HUP slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                killproc -HUP slurpd
                RETVAL=$?
            fi
        fi
        ;;
    *)
        echo "Usage: $0 start|stop|restart|status}"
        exit 1
esac

exit $RETVAL
```

If you use this file with a version of OpenLDAP that you compiled yourself, you will need to change the paths of the executable and config files to reflect your installation. Next you will need to add run level links to this file. I startup the OpenLDAP server in runlevels 2,3,4 and 5 and kill it in run levels 0,1 and 6. Use the following commands from /etc/rc.d/init.d to create the appropriate symbolic links:

Example 4-3. Create The Proper Symlinks

```
ln -s ldap /etc/rc.d/rc0.d/K31ldap
ln -s ldap /etc/rc.d/rc1.d/K31ldap
ln -s ldap /etc/rc.d/rc2.d/S79ldap
ln -s ldap /etc/rc.d/rc3.d/S79ldap
ln -s ldap /etc/rc.d/rc4.d/S79ldap
ln -s ldap /etc/rc.d/rc5.d/S79ldap
ln -s ldap /etc/rc.d/rc6.d/K31ldap
```

This will start ldap before sendmail and kill it after sendmail so that any unprocessed mail can get through before the LDAP server goes down.

4.3.2. slapd.conf

You will need to make the following changes to the Open LDAP configuration file. This is where some of the LDAP learning curve starts to hit, so pay attention. The slapd.conf file will be located in /usr/local/etc/openldap/ if you compiled OpenLDAP with defaults or /etc/openldap if you used the RPM. The modified file should look like the following example minus the notes. Paths should be changed depending on your LDAP installation.

> Note: Replace all data inside <> (including <>) to reflect your information.

Example 4-4. Listing of slapd.conf

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /usr/local/etc/openldap/slapd.at.conf
include         /usr/local/etc/openldap/slapd.oc.conf
schemacheck     off
#referral       ldap://ldap.itd.umich.edu
```

```
 pidfile         /usr/local/var/slapd.pid
 argsfile        /usr/local/var/slapd.args


######################################################
 # ldbm database definitions

######################################################

 database        ldbm

 Note: The suffix is the called the Base DN by most LDAP utilities.
 It is the root of the LDAP tree, all data branches from the Base
 DN.  Two of the most common Base DN's are the domain name and the
 organization name plus the country code.  You may choose whichever
 you like, but all subsequent entries must reflect your choice.

#suffix       "dc=<your domain name>, dc=<your domain extension ie com, net, org>"
 suffix          "o=<your organization name>, c=<your country code ie US>"
 directory       /usr/tmp

 Note: The rootdn is the default ldap user created when the server
 starts and is not in the database itself.  This is the user that
 you will use to manage the LDAP database and is commonly called
 the Bind DN by many tools and utilities.  The rootpw is also known
 as the Bind Password.  This file should be protected, due to the
 fact that control of your LDAP database lies in these two lines.
 You may encrypt this entry, see the man page for slapd.conf.

 #rootdn          "cn=root, dc=<your domain name>, dc=<your domain extension>"
 rootdn          "cn=root, o=<your organization name>, c=<your country code>"
 rootpw          secret
 # cleartext passwords, especially for the rootdn, should
 # be avoided.  See slapd.conf(5) for details.

 Note: The following are Access Control Lists that I have on my
```

```
LDAP server, these were modified from example ACLs on the OpenLDAP
Faq-O-Matic.  I have an Administrators group created and I limit
access to passwords for user entries.  Keep in mind that you may
need to add this section after you have added the initial entries
into the LDAP server.  These ACLs may generate errors on startup
if the entries they protect are not in the LDAP database.

index default pres,eq,approx,sub,none
lastmod on

defaultaccess read
access to dn="cn=Administrators,o=<your organization name>,c=<your country code>"
  by dnattr=member selfwrite
  by * none
access to dn="*,o=<your organization name>,c=<your country code>"
  by self write
  by dn="cn=root,o=<your organization name>,c=<your country code>" write
  by * read
access to attr=userPassword
  by self write
  by dn="cn=root,o=<your organization name>,c=<your country code>" write
  by * compare
```

4.3.3. Initial LDIF File and ldapadd

Next we will add some initial entries to the LDAP server. I suggest creating an LDIF file and loading the contents into the database using the ldapadd utility. This is the content of my initial LDIF file. If you have suggestions for additional fields and classes in order to help with compatibility with various LDAP clients such as MS Outlook or Netscape Navigator, please let me know. This is a bit minimal and since the scope of this document is to replace an MS Exchange server, we would like to make our LDAP entries as feature rich as possible. Remember to replace all information inside <> with your own. Multiple entries of same type within a dn (distinguished name) will simply return multiple results.

You may use multiple entries to not options such as a state abbreviation and a state spelled out for the 'st' field.

Example 4-5. LDIF File

```
 dn: o=<your organization name>, c=<your country code>
 dc: <your domain name>
 dc: <your domain extension>
 o: <your organization name>
 l: <your city>
 st: <your state or province mail abbreviation>
 st: <your state or province, spelled out>
 postalcode: <in the US, your zip code>
 postofficebox: <your street address>
 c: <your country code>
 telephonenumber: <organization phone number>
 objectclass: top
 objectclass: organization

Note: This is a group of names that I use for ACLs and other various management tasks
 dn: cn=Administrators, o=<your organization name>, c=<your country code>
 cn: Administrators
 objectclass: groupofNames
 objectclass: top
 member: cn=<First User>, o=<your organization name>, c=<your country code>
member: cn=<Second User>, o=<your organization name>, c=<your country code>

 dn: cn=<First User>, o=<your organization name>, c=<your country code>
 cn: <First User>
 sn: <User>
 givenname: <First>
 objectclass: person
 objectclass: uid
 objectclass: organizationalPerson
 objectclass: top
 locality: <city>
 st: <state or province>
```

```
mail: <email address>
title: <title>
postofficebox: <street address>
postalcode: <zip code>
countryname: <your country code>
telephonenumber: <your phone number>
o: <your organization name>
xmozillanickname: <First>
xmozillausehtmlmail: TRUE
pagerphone: <pager number>
uid: <User id, this is the username that is used in the email address>

dn: cn=<Second User>, o=<your organization name>, c=<your country code>
cn: <Second User>
sn: <User>
givenname: <Second>
objectclass: person
objectclass: uid
objectclass: organizationalPerson
objectclass: top
locality: <city>
st: <state or province>
mail: <email address>
title: <title>
postofficebox: <street address>
postalcode: <zip code>
countryname: <your country code>
telephonenumber: <your phone number>
o: <your organization name>
xmozillanickname: <Second>
xmozillausehtmlmail: TRUE
pagerphone: <pager number>
uid: <User id, this is the username that is used in the email address>
```

Once this has been written to a file <yourldiffile>.ldif you may import it using the the ldapadd utility. First make sure your LDAP server has been started. If you receive errors about the ACLs, comment them out and restart the LDAP server.

Next change directories to the directory containing the ldap file you just created and run the ldapadd utility. You will give the -D argument with the Bind DN or rootdn in quotes, the -W argument and then stream the <yourldiffile>.ldif into the ldapadd utility.

```
ldapadd -D “cn=root,o=<your organization name>,c=<your country code>” -W
< <yourldiffile>.ldif
```

You will be prompted for the rootpw (Bind Password). Enter the password and the utility should dump you to a command prompt if no errors are encountered. You may want to look over the man page for ldapadd.

4.3.4. ldappasswd: Setting user passwords

Use the ldappasswd utility to set user passwords. For the two entries we just added, to give them passwords, we will type the following:

```
ldappasswd -D “cn=root,o=<your organization name>,c=<your country code>”
-w <rootpw> -t “cn=<First User>,o=<your organization name>,c=<your
country code>”
```

This will prompt us for a new password, which will be stored in the LDAP database as a UNIX crypt password. There are other options for encryption, but they may require modification to the cyrus server configuration. This is also the reason we use ACLs on the userPassword entry, there are other safeguards built into OpenLDAP, please read the documentation on their website and the man pages. If you find a more secure way to set this up, please let me know so that we can add it in here.

```
ldappasswd -D “cn=root,o=<your organization name>,c=<your country code>”
-w <rootpw> -t “cn=<Second User>,o=<your organization name>,c=<your
country code>”
```

Again you will be prompted for the new password. Also, you should notice the -t argument, this is the target dn. In other words, this is the dn that we are changing the password for.

4.4. pam_ldap Configuration Files

4.4.1. /etc/ldap.conf

You will need to create a file, /etc/ldap.conf, if you built your own pam_ldap or edit the existing one if you installed with the RPM. There are really only 2 lines that must be configured. The host and base entries.

Example 4-6. ldap.conf

```
host 127.0.0.1

base o=<your organization name>,c=<your country code>
```

Note the lack of spaces between the comma and the c=. For some reason, OpenLDAP likes this method better when being queried. I am not sure why, but I have had LDAP look up failures when querying the server with spaces between the commas and items. Also note that the host entry will allow you to use an LDAP server on another machine for pam authentication. The base entry must reflect your suffix entry (Base DN) in the slapd.conf file.

4.4.2. imap and pop files

You will need to configure the imap and pop files in the /etc/pam.d directory so that the cyrus imap server will use the pam_ldap module. Both files will need to look like the following:

Example 4-7. imap & pop files

```
#%PAM-1.0
auth       required    /lib/security/pam_ldap.so
account    required    /lib/security/pam_ldap.so
```

4.5. Cyrus IMAP configuration

4.5.1. imapd.conf

The imapd.conf file should reside in /etc/. It is a rather simple file. The following is a standard imapd.conf where the imap user files and mailboxes are under /var/. The admins entry is the real important one. You must have a corresponding user in the LDAP database for the admins. The admins entry may contain a space separated list of users, who all have entries and passwords in the LDAP database.

Example 4-8. The imad.conf

```
configdirectory: /var/imap
partition-default: /var/spool/imap
admins: <First User>
sasl_pwcheck_method: PAM
```

The admins entry makes reference to a user that will be setup in the Open LDAP initial entries section of this document. You will need to replace <First User> with the actual name of a user in your LDAP database. This user or these users will have permission to create, delete and modify mail boxes and accounts within cyrus.

There are many other options for the ldap.conf file, If you compiled your own pam_ldap and are interested, please download the nss_ldap libraries from http://www.padl.com and examine the ldap.conf contained in the archive. There are also some other options explained in the ldap.conf file included with the RPM.

4.5.2. imap directories under /var

> Note: if you used RPMs, most of this should be done for you, in fact the mkimap utility listed below is not included in the RPM.

First create an imap directory under /var and /var/spool. To do this execute the following commands:

Example 4-9. Creating Required Directories

```
cd /var

mkdir imap

chown cyrus imap

chgrp mail imap

chmod 750 imap

cd /var/spool

mkdir imap

chown cyrus imap

chgrp mail imap

chmod 750 imap
```

Change directories to the tools directory under the cyrus-imap source directory. There should be an executable named mkimap. su to the cyrus user, su cyrus, and type ./mkimap. Change directories to /var/imap. You will need to set the sync flag on several files and directories. This is done by typing the following commands:

Example 4-10. Setting the Sync Flag

```
cd /var/imap

chattr +S . user quota user/* quota/*
```

```
chattr +S /var/spool/imap /var/spool/mqueue
```

4.5.3. setting up logging for cyrus

If you want to generate a log file for the imap server add the following line to /etc/syslog.conf and restart the syslog daemon by typing /etc/rc.d/init.d/syslog restart.

Example 4-11. Log Settings

```
local6.debug /var/log/imapd.log
```

Create the log file for imapd by typing touch /var/log/imapd.log. Next add the daemon user to the mail group in the file /etc/group.

4.5.4. cyradm: adding mail users

The cyradm utility is used to manage mailboxes on the cyrus server. This utility is scriptable in Tcl. If you are familiar with Tcl you may want to write a script to add many users at once. There are also some examples in the doc directory of the cyrus-imap archive. To simply add a user with the command line, you must first log into the cyrus server as an admin defined in the imapd.conf file. Type the following:

Example 4-12. Add a User

```
cyradm -u <First User> localhost
```

You will be prompted for a password, make sure that the LDAP server is up and running and that the user has an entry with a password. Enter the password and you should be given a prompt: >. At the prompt you can type help for a list of commands. To create a mailbox type the following:

Example 4-13. Create a Mailbox

```
>cm user.<uid>
```

<uid> should be replace with the uid entry for the user you are creating the mailbox for. For example, if you are creating a mailbox for the <First User> account and the email address will be fuser@mydomain.com, then the uid field in the LDAP database should be fuser. The command you should type at the cyradm prompt would be >cm user.fuser. If you then do an >lm you should see user.fuser listed. For more information on the cyradm utility, please see the man page. There are more options, including the creation of public folders and ACLs that may be used in conjunction with IMAP4 accounts. If you intend to use POP3 accounts please read the next section.

4.5.5. POP3 accounts

If your users will be using POP3, you must create mailboxes, as described above. After doing so, you must create a directory under /var/imap/log with the users uid. For example, if <First User> (uid: fuser) wants POP3 access, then we would do the following:

Example 4-14. Creating mailboxes

```
cd /var/imap/log

mkdir fuser

chown cyrus fuser

chgrp mail fuser
```

```
chmod 700 fuser
```

As long as the LDAP server is running, and the cyrus imap server and `pam_ldap` are configured properly, <First User> should be able to log on using an IMAP or POP3 client and check their mail.

4.6. Sendmail Configuration

There are several parts to the sendmail configuration that must be used.

4.6.1. Sendmail.mc

Below is a sendmail.mc file with notes:

Example 4-15. sendmail.mc file

```
divert(-1)dnl
#
#  This file contains definitions for mail.mydomain.com
#
divert(0)dnl
include(`/usr/lib/sendmail-cf/m4/cf.m4')
VERSIONID(`@(#)mailserver.mc     1.0 (mydomain.com) 5/1/97')
OSTYPE(linux)
DOMAIN(generic)
dnl Note: The following feature is an aliases file that allows
dnl sendmail to recognize mail coming in to several host names
dnl for this machine
FEATURE(`use_cw_file')
define(`confCW_FILE',`/etc/sendmail.cw')
FEATURE(nouucp)
dnl Note: The virtusertable feature allows you to create an aliases
dnl file for users that only have cyrus accounts or to forward mail to
```

```
dnl another address not handled by this server. The access_db feature
dnl allows you to control which hosts and addresses may relay mail
dnl using this machine. For more information on this and other
dnl anti-spam features, see http://www.sendmail.org/m4/anti-spam.html (http://www.sendmail.org/m4/anti-spam.html).
FEATURE(‘virtusertable’, ‘hash /etc/mail/virtusertable’)
FEATURE(‘access_db’, ‘hash /etc/mail/access’)
dnl Note: This allows all relayed messages to appear as if they
dnl are coming from user@mydomain.com instead of user@host.mydomain.com
MASQUERADE_AS(mydomain.com)
dnl Note: This is probably one of the most important entries,
dnl LUSER_RELAY tells sendmail to send mail for users that do not have
dnl local system accounts on this machine to the cyrus mailer on
dnl localhost. Without this entry, all mail addressed to cyrus users
dnl would bounce as user unknown.
define(‘LUSER_RELAY’, ‘cyrus:localhost’)
dnl Note: rbl uses the Realtime Blackhole List database to keep
dnl known spammers from accessing your mail server, for more
dnl information, please see http://www.sendmail.org/m4/anti-spam.html.
FEATURE(rbl)
dnl Note: This feature allows users mail to be forwarded with a
dnl .forward file in the users home directory, this is only for users
dnl with system accounts, for cyrus only users, use the virtusertable
dnl feature.
FEATURE(‘redirect’)
dnl Note: This uses procmail instead of the old mail executable for
dnl local delivery.
FEATURE(‘local_procmail’)
dnl Note: These are the mailer definitions, this allows sendmail to
dnl use smtp to deliver outgoing mail, cyrus for imap and pop3 users
dnl and procmail for local system account delivery (root).
MAILER(smtp)
MAILER(procmail)
MAILER(cyrus)
```

You may need to add or modify features for your particular needs. When you are done generate the sendmail.cf file by typing the following command: m4 sendmail.mc > sendmail.cf

Both of these files should reside in /etc.

4.6.2. Directing system account mail to the cyrus mailer

On my mail server I have a system account setup for myself which has the same username that I use for my mail address. Sendmail was delivering my mail to the local account rather than the cyrus account. I did not want to use cyrus for all my system accounts, especially since my cyrus imap server was using pam_ldap to authenticate users. In my /etc/sendmail.cf file that I output using the /etc/sendmail.mc file in the above section, I added the following under the Class definition section:

Example 4-16. Sendmail Class Definitions

```
# class C: names that should be sent to cyrus
CC <First User>
```

In the Parse1 section of Ruleset 0 I added the following 2 lines to the #short circuit local delivery so forwarding works sub-section:

Example 4-17. Short Circuiting Delivery (Original)

```
R$=C < @ $=w . >        $#cyrus $: @ $1         special <First User> rule

R$=C < @ $=w . >        $#cyrus $: $1           special <First User> rule
```

So now my #short circuit local delivery so forwarding works sub-section of the Parse1 section looked like the following:

Example 4-18. Short Circuiting Delivery (New)

```
# short circuit local delivery so forwarded email works

R$=C < @ $=w . >        $#cyrus $: @ $1         special <First User>rule

R$=C < @ $=w . >        $#cyrus $: $1          special <First User>rule

R$=L < @ $=w . >        $#local $: @ $1        special local names

R$+ < @ $=w . >         $#local $: $1          regular local names
```

I also had to add the following 2 lines to the #handle locally delivered names sub-section of the Parse1 section:

Example 4-19. Handling Locally Delivered Names (Original)

```
R$=C @ $=w              $#cyrus $: @ $1               special <First User>rule

R$=C                    $#cyrus $: @ $1               special <First User>rule
```

So now the #handle locally delivered names sub-section of the Parse1 section looked like the following:

Example 4-20. Handling Locally Delivered Names (New)

```
# handle locally delivered names

R$=C @ $=w              $#cyrus $: @ $1               special <First User>rule

R$=C                    $#cyrus $: @ $1               special <First User>rule

R$=L                    $#local $: @ $1               special local names

R$+                     $#local $: $1                 regular local names
```

There is a way to add these rules and class to the sendmail.mc file rather than the gory additions to the sendmail.cf file, if you would like to study up on sendmail and send me the lines for the sendmail.mc file, I will gladly replace this section with them. For every user that needs to have mail delivered to cyrus instead of locally, add their user id to the class definition line. This list should be single-space separated. This inelegant method must be administered every time you rebuild the sendmail.cf file from the sendmail.mc file.

4.6.3. Restarting and monitoring sendmail

Every time your /etc/sendmail.cf changes you must restart sendmail to have the changes take effect. Do this by using the sendmail startup script in /etc/rc.d/init.d/ by typing /etc/rc.d/init.d/sendmail restart . You may also check for errors in the sendmail logfile: /var/log/maillog. I usually run tail -f /var/log/maillog when I am testing sendmail configurations and open the log file in an editor only when I am looking for past errors or information.

4.7. GQ LDAP GUI setup

The easiest way to add single LDAP entries is through gq. Start gq on a linux box running X-Windows.

4.7.1. Adding a Server

Choose Preferences from the File Menu. Select the Servers Tab and click on the New button. You should have a new window that has 2 tabs with the General tab currently selected. Enter a friendly name in the Name field, this will appear in your gq server list. Enter the address of the LDAP server in the LDAP host field. You should leave the LDAP Port alone. In the Base DN field enter the suffix entry from your slapd.conf file. Next select the Details tab, enter the rootdn entry from the slapd.conf file in the Bind DN field and the rootpw entry in the Bind Password field. Click OK on the New Server window and Click OK on the Preferences window.

4.7.2. Testing

The browse mode is the easiest to use. Choose Browse from the Mode menu. You should see the Name of the new server you added. Click on it and it should expand. If you have added the organization and two users as shown in the OpenLDAP configuration section, you should be able to expand the tree control and see all of the entries.

Chapter 5. Adding Mail Accounts

5.1. Creating an LDAP entry

5.1.1. Creating from a template

Use gq in Browse mode to connect to your LDAP server, expand the server until you see your user entries. Right click (button 3) over a user entry and select the use as template option from the menu. This will give you a blank entry to your right. Simply fill out the blank fields and click the Apply button at the bottom. You may want to open another instance of gq to use as a visual template to make sure that everything matches. If you make a mistake, make sure it is not in the dn, entries can get ’lost’ if the dn is incorrect.

5.1.2. Lost entries

If an entry is lost, you will need to manually delete the entry with the ldap command line utilities . A more extreme solution may be needed: export the contents of the database as an LDIF file, manually repair it, empty the database file with an editor and restore the database using ldapadd and the LDIF file.

5.1.3. Adding a password

>From the command line, use the ldappasswd utility to create a password entry for the user. Type the following, assuming we just added <New User> to the LDAP database using gq:

```
ldappasswd -D “cn=root,o=<your organization name>,c=<your country code>”
-w <rootpw> -t “cn=<New User>,o=<your organization name>,c=<your
```

country code>”

You will be prompted for the password for the new user and asked to type it again.

5.2. Creating the cyrus mailbox

5.2.1. cyradm

>From the command line, log on to the cyrus server as the user listed in the admins entry of the imapd.conf file

cyradm -u <First User> localhost

After logging in with the password in the LDAP database for <First User>, create the mailbox with the new users uid LDAP entry, at the cyradm prompt. (Assuming the new user, <New User>, has a email address of nuser@mydomain.com and a uid entry in the LDAP database of nuser)

Example 5-1. Creating the mailbox

```
>cm user.nuser

>lm

INBOX          user.nuser

user.fuser    user.suser

>lam user.nuser

nuser lrswipcda
```

The previous lines create the mailbox, list the mailboxes to make sure that it worked and then list the permissions on the new mailbox. Next, exit the cyradm utility.

```
>quit
```

5.2.2. (Optional) Adding POP3 access

To allow the user to access their mail using a POP3 client, add a directory to /var/imap/log with the user's corresponding uid. Make sure that the permissions are correct. For example, <New User> wants POP3 access:

Example 5-2. Commands to Add POP3 Access

```
cd /var/imap/log

mkdir nuser

chown cyrus nuser

chgrp mail nuser

chmod 750 nuser
```

5.3. Testing the new account

5.3.1. Sendmail

Try sending the new user a test message, if you do not get a user unknown error, and the sendmail log files have a line containing the user's email address and a Status=Sent then you should be OK in that area. If not then check the mail box with cyradm, most likely cyrus is not connecting to the LDAP server or the mailbox was not created.

5.3.2. IMAP

Use the imtest utility in the imtest directory of the cyrus-imap-<version> source code directory, or /usr/bin/imtest, if you used the RPM. For example, testing <New User> with a uid of nuser

```
    [root@localhost bin]# imtest -u nuser -m login -p imap -v localhost
    S: * OK mail.mydomain.com Cyrus IMAP4 v1.6.13 server ready
    C: C01 CAPABILITY
    S: * CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ NAMES-
PACE UIDPLUS X-NON-HIERARCHICAL-
RENAME NO_ATOMIC_RENAME AUTH=PLAIN AUTH=CRAM-MD5 UNSE-
LECT X-NETSCAPE
    S: C01 OK Completed
```

When prompted enter the password in the LDAP database for the <New User> entry

```
Password:
+ go ahead
L01 OK User logged in
Authenticated.
Security strength factor: 0
To logout simply type .logout
. logout
* BYE LOGOUT received
. OK Completed
Connection closed.
[root@localhost bin]#
```

If you receive errors, look in the cyrus logfiles and the ldap logfiles. Most likely the problem is with the cyrus mailbox or the LDAP entry.

5.3.3. POP3

If you set up a POP3 directory for <New User> you can test it by typing the following:

Example 5-3. Test New POP3 User

```
[root@localhost root]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK mail.mydomain.com Cyrus POP3 v1.6.13 server ready
```

Type the following line, assuming that <New User> has a uid of nuser in the LDAP database and a cyrus mailbox and directory were created using that uid.

Example 5-4. Testing New POP3 User (cont.)

```
USER nuser
+OK Name is a valid mailbox
```

Type the following line, assuming that the password in the LDAP database for <New User> is imnew

Example 5-5. Testing New POP3 User (cont.)

```
PASS imnew
+OK Maildrop locked and ready
```

Type the following to quit:

Example 5-6. Testing New POP3 User (cont.)

```
QUIT
+OK
Connection closed by foreign host.
[root@localhost root]#
```

Chapter 6. Conclusion

6.1. Today

This document should get you through the basics of installation and configuration for a server that will handle SMTP, IMAP4 and POP3 as well as providing a company wide LDAP directory of email addresses. This is the most basic of services that we hope to eventually provide.

6.1.1. Benefits

My experience with both UNIX and Microsoft systems has led me to the conclusion that I can no longer work on Microsoft systems. While Exchange Server offers tighter integration than this collection of software and easier (GUI) system administration, I'd rather not be awakened by a pager in the middle of the night or be completely unable to determine the cause of problems. I've never been more embarrassed in my professional life than telling a manager or client, "I don't know why it broke or how I fixed it.". The peace of mind that I have gained with this architecture is priceless. Troubleshooting and debugging leads to concrete answers that can be fixed, given the time.

6.1.2. Drawbacks

The biggest drawback to this system is security. This is simply a matter of someone getting a working system. The OpenLDAP server is capable of SSL enabled LDAP connections, most clients are capable of SSL encrypted IMAP4, POP3 and LDAP sessions. The pam_ldap module supports SSL using the Netscape C SDK precompiled library. I believe Cyrus supports SSL IMAP and POP3 sessions. As long as your data does not pass over public networks you are somewhat safe. However please keep in mind when evaluating this system, that security should remain a top priority. If anyone has any information on how they may have locked down their connections, please let me know.

Lack of server side calendar coordination and client mailing list management. These are the 2 major drawbacks as far as functionality is concerned. I will be looking at solutions as I come across them, but any help in this area will also be greatly appreciated.

The Sympa Mailing List Manager at http://listes.cru.fr/sympa/ looks promising. Complete LDAP support is not available yet. I am primarily concerned with using LDAP to manage lists, list owners and subscribers, perhaps by having a Sympa User LDAP entry that has an objectclass entry for sympa. The list entry would then contain a list of users and the list manager. You could do this without modifying users. The actual messages could be stored via Sympa's default storage methods.

Administration is also a big concern. OpenLDAP is working on building LDIF and schema files that will allow you to load most of the configurations that popular clients look for when utilizing LDAP servers. This project is a bit slow and scattered however. If you run into any LDAP related issues that involve the schema or ACLs, please let me know so that we can update the information, and hopefully help out the OpenLDAP project. Also, administration utilities for converting mailboxes from other systems and a single interface for setting users up within this architecture would be quite welcome.

I am currently building a Perl CGI script that will allow web-based user management. Currently it modifies the configuration file for the script, adds users, and partially modifies users. I'll be finishing the scripts basic functionality before Febuary 2000. The current problems are security, and POP3. In order to enable pop3 a directory must be created on the filesystem with owner cyrus and group mail. The directory that this must be created in is owned by cyrus and mail. As most web servers run as nobody/nobody, this creates a permissions problem. The most likely solution for this problem will be to turn the script into an inetd controlled server, like linuxconf's web interface or Samba's swat. This way you may use host.allow and hosts.deny to control access. The only drawback, is that using a web server, you can use the CGI through an SSL enabled connection. SSL is a definate plus since you will be passing passwords around.

6.2. Tomorrow

As Open Source solutions present themselves, I hope to integrate them into this system, allowing for more and more 'groupware' functionality. Hopefully one day, this architecture will surpass the most expensive commercial solutions available in terms of stability, scalability and functionality. Some of the issues that I hope to bring to this document are:

- Client configuration
- Security
- Client Managed Mailing Lists
- Document Sharing
- Calendaring
- Administration Utilities and Methods
- Backing up Mailboxes and data

If you have any information that you feel will help the creation of a superior Open Source groupware platform, please let me know.

Chapter 7. Acknowledgements

Many thanks to the developers and maintainers of the following projects,

Sendmail

OpenLDAP

GQ

Cyrus IMAP Server

Cyrus SASL Libraries

PAM

PAM LDAP module

Mozilla

Please help these projects in any way you can. Their dedication has enabled many an administrator to sleep soundly at night (or to sleep at all).