Setting up postfix+ldap+courier+squirrelmail

Preparations

First of all, my believe that to successfully manage a system, you must use rpm to distribute software - compiling and installing from source may work at the beginning, and in small scale, but you quickly get bogged down and the system becomes unmanageable.

At this document i will guide you how to install postfix+ldap+courier+squirrelmail on a RedHat 7.3 system. I will provide additional detaild for doing the same on RH 9, since the sasl configuration will be different due to version used in each distro. And TRUST ME, if you are using 7.3 distro, DO NOT try to include your own, updated openldap or sasl packages - this will give you such a headache, that after 2 days of work you will say that it is not worth it - these libs are deeply embedded into redhat's base, you have to recompile half the distro to use them successfully.

To build my postfix package, i get the src.rpm from http://postfix.wl0.org/, and rebuild it's spec file to include `ldap=1`, `sasl=1`, `tls=1`, `pcre=1`. To do that, you will need to install the src.rpm, cd into */usr/src/redhat/SOURCE*,

```
export POSTFIX_LDAP=1
export POSTFIX_SASL=1
export POSTFIX_TLS=1
export POSTFIX_PCRE=1
```

then run `sh make-postfix.spec`. you also need to install *libtool* and *gcc-c++* before building the package (`rpm -ba SPECS/postfix.spec`).

To build my CourierImap packages, get the tar.gz from http://www.courier-mta.org/download.php. To build the rpm from the tar gz, some magic is needed, since *you need to build it as a non-root user*. i usually choose a non-root user, `chown $1 -R /usr/src/redhat`, and then `rpm -ta filename`.

All these packages are available here : TODO !!! 

Installation

Now, install the *openldap-servers*, *openldap-clients* and *nss_ldap* packages from your RedHat distro.

OpenLDAP

Your server configuration starts with setting up your LDAP server with initial schema. Copy the authldap.schema and qmail.schema files to /etc/openldap/schema. Now edit your /etc/openldap/slapd.conf to include the following lines (sample full file is also available) :

```
# Postfix / Qmail
include         /etc/openldap/schema/qmail.schema
# courier IMAP
include         /etc/openldap/schema/authldap.schema

# default permissions on the database
access to dn="" by * read
access to *
       by self write
       by users read
       by anonymous read

access to attr=userpassword,clearpassword,ldappassword
        by self write
        by anonymous auth
        by * none

# You may want to enable debuggind while setting up the server.
# The messages are .info level, so change the default syslog settings to see
them
# loglevel 4

suffix          "dc=example,dc=com"
rootdn          "cn=Manager,dc=example,dc=com"
rootpw          alex

index cn eq
#faster mail queries I hope
index mail eq
index givenname eq
index uid eq
#address book lookups
index sn eq
#squirrelmail LDAP plug-in uses this
index objectClass eq
```

Start the ldap server.

Create an initial schema ldif file, `ldap-initial.ldif` :

```
# example, com
dn: dc=example, dc=com
objectClass: top
objectClass: organization
o: my company
description: top level of directory

# mailaccounts, example, com
dn: ou=mailaccounts, dc=example, dc=com
objectClass: top
objectClass: organizationalunit
```

```
ou: mailaccounts
description: people with mailaccounts at my company.
```

Add it's contents to the server :

```
ldapadd -D "cn=manager,dc=example,dc=com" -w alex -x -f ldap-initial.ldif
```

the output should be :

```
adding new entry "dc=example, dc=com"

adding new entry "ou=mailaccounts, dc=example, dc=com"
```

Create a user template file, `user_template.ldif`:

```
dn: uid=UID,ou=mailaccounts,dc=example,dc=com
uid: UID
cn:  UID
uidNumber: 1001
gidNumber: 1001
mail: UID@example.com
mailHost: mail.example.com
homeDirectory: /var/imap/example.com/UID
mailMessageStore: /var/imap/example.com/UID/Maildir
mailbox: UID/Maildir/
objectClass: qmailuser
objectClass: couriermailaccount
accountStatus: active
```

Create preprocessing script for user creation, process_user :

```
#!/usr/bin/perl
($userid)=@ARGV;
while ($line=<STDIN>){ $line =~s/UID/$userid/; print $line;}
chmod +x process_user
```

this script will read input from STDIN, and replace UID with the first parameter given to it.

Let's add a test user:

```
cat user_template.ldif  | ./process_user test | ldapadd -D
"cn=Manager,dc=example,dc=com" -w alex -x
```

the output should be :

```
adding new entry "uid=test,ou=mailaccounts,dc=example,dc=com"
```

Set the password for the user:

```
ldappasswd -D "cn=Manager,dc=example,dc=com" -w alex -s testpass -x
```

```
uid=test,ou=mailaccounts,dc=example,dc=com
```

The output should be :

```
Result: Success (0)
```

Now, install postfix and courier rpms you have compiled :

Postfix

```
rpm -i postfix-2.0.6-3.pcre.sasl1.tls.rh73.i386.rpm
/mnt/atlas/misc/postfix/courier-imap-ldap-1.7.1-1.7.3.i386.rpm
/mnt/atlas/misc/postfix/courier-imap-1.7.1-1.7.3.i386.rpm
```

Let's see if the default config we got is different from postfix default :

```
# postconf  -n
alias_database = hash:/etc/postfix/aliases
alias_maps = hash:/etc/postfix/aliases
command_directory = /usr/sbin
config_directory = /etc/postfix
daemon_directory = /usr/libexec/postfix
debug_peer_level = 2
mail_owner = postfix
mailq_path = /usr/bin/mailq.postfix
manpage_directory = /usr/share/man
newaliases_path = /usr/bin/newaliases.postfix
queue_directory = /var/spool/postfix
readme_directory = /etc/postfix/README_FILES
sample_directory = /etc/postfix/samples
sendmail_path = /usr/sbin/sendmail.postfix
setgid_group = postdrop
unknown_local_recipient_reject_code = 450
```

we see that some stuff is customised to redhat's distro, but the only thing that is important is that unknown_local_recipient_reject_code is set to 450 - after we get everything working, we will need to comment out this setting - the default is 550.

Now, we are going to configure postfix to deliver to virtual mailboxes using ldap. The delivery will be done to maildirs, under the user vmail.

Create user vmail:

```
useradd vmail -d /var/lib/courier
```

Create the store directory, and the domain directory :

```
mkdir /var/imap/example.com -p
chown vmail:vmail /var/imap/example.com -R
```

set postfix to deliver under user vmail, and set it to query ldap for local deliveries

```
postconf -e myhostname=mail.example.com mydomain=example.com
default_privs=vmail
```

add by hand to `/etc/postfix/main.cf`:

```
virtual_mailbox_base = /var/imap/example.com
virtual_mailbox_maps = ldap:ldapsource
virtual_gid_maps = static:<UID OF VMAIL USER>
virtual_uid_maps = static:<GID OF VMAIL USER>
virtual_minimum_uid = 500
virtual_mailbox_domains = example.com
virtual_result_attribute = mailbox
virtual_maildir_extended = yes

ldapsource_timeout = 10
ldapsource_server_host = localhost
ldapsource_search_base = ou=mailaccounts,dc=example,dc=com
ldapsource_server_port = 389
ldapsource_domain = example.com
ldapsource_query_filter = (&(mail=%s)(accountstatus=active))
ldapsource_result_attribute = mailbox
ldapsource_bind = no
```

Start postfix, and try to send an email to the `test@example.com` user :

```
echo test | sendmail test@example.com
```

check the `/var/log/maillog` for errors.

If you get ldap-related errors, set loglevel to 4 in `/etc/openldap/slapd.conf`, change - to

```
#*.info;mail.none;authpriv.none;cron.none                /var/log/messages
*.*;mail.none;authpriv.none;cron.none                 /var/log/messages
```

and restart syslog, and look at `/var/log/messages` for ldap errors

Courier-IMAP

Let's configure courier for pop3 and imap access now :

Install the `courier` and `courier-ldap` rpms.

edit `/usr/lib/courier-imap/etc/authdaemonrc`

```
authmodulelist="authldap"
```

enter this to `/usr/lib/courier-imap/etc/authldaprc` - delete all the rest

```
LDAP_SERVER             localhost
LDAP_PORT               389
LDAP_BASEDN             ou=mailaccounts,dc=example,dc=com
LDAP_AUTHBIND           1
LDAP_TIMEOUT            5
LDAP_MAIL               mail
LDAP_DOMAIN             example.com
LDAP_GLOB_GID           vmail
LDAP_GLOB_UID           vmail
LDAP_HOMEDIR            homeDirectory
LDAP_MAILDIR            mailDir
LDAP_FULLNAME           cn
LDAP_DEREF              never
LDAP_TLS                0
```

start courier, and telnet to port 110. type „user test“, „pass testpass“ - this should produce no error.

Setting up SMTP authentication:

Because on redhat 7.3 the `sasl` version is 1.5, and it doesn't support `authdaemon` which is available at version 2.0, we will have to setup a less secure configuration - meaning, to un-chroot the smtp daemon.

```
vi /etc/postfix/master.cf :
```

on the line of

```
smtp    inet
```

set chroot to „n“

activate sasl authentication for postfix in `/etc/postfix/main.cf`:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
```

the default relaying blocking is:

```
smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
```

this will only allow users from mynetworks to send mail outside the domains postfix is aware of. my config is as follows :

```
smtpd_recipient_restrictions = permit_sasl_authenticated, permit_mynetworks,
 check_recipient_access hash:/etc/postfix/protect-accounts,
reject_unauth_destination
```

I permit also sasl authenticated users. I also protect my mailinglist accounts inside my mail server (for example, all@example.com), so people not on my network, or authenticated with sasl, will not be able

to send mail to the protected address. The protect-accounts should look like this:

```
all@example.com     REJECT
another_address@example.com     REJECT
```

If you do not need this functionality, you can remove `check_recipient_access hash:/etc/postfix/protect-accounts`

let's configure sasl

Now, let's configure sasl, so it would do the authentication:

configure your `/etc/ldap.conf`:

```
host 127.0.0.1
base dc=example,dc=com
```

create a file `/usr/lib/sasl/smtpd.conf`, containing :

```
pwcheck_method: PAM
```

create a file `/etc/pam.d/smtp` :

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

Miscellaneous

to delete an account, use

```
ldapdelete -D "cn=Manager,dc=example,dc=com" -w alex -x
uid=testuser,ou=mailaccounts,dc=example,dc=com
```