Mail-Server Setup with DragonFlyBSD, Postfix, Dovecot and Dspam

Michael Neumann (mneumann@ntecs.de)

Version 1.1 (2010-08-17)

ChangeLog

- 1.1: Added dspam_stats and dspam_clean weekly periodic script

1 Overview

This howto describes the setup of a complete mail system on DragonFly. It is suited for small mail setups only and is not optimized for performance.

2 Installation

The host system this installation is running on is DragonFlyBSD:

```
# uname -a
DragonFly kvmdragon.ntecs.de 2.6-RELEASE DragonFly v2.6.3.39.g0a997-
    RELEASE #2: Wed Aug 11 19:05:18 CEST 2010     root@kvmdragon.ntecs.de
    :/usr/obj/usr/src/sys/KVM  i386
```

We need the following packages from pkgsrc:

- databases/postgresql84 (8.4)
- mail/postfix (2.7.1)
- mail/dovecot (1.2.13)
- wip/dovecot-antispam (1.2.90.1)
- mail/dspam (3.8.0)
- mail/postgrey (1.33)

Lets install all of them before doing any specific configuration. In /usr/pkg/etc/mk.conf add the following lines which specifies options for the packages.

```
PKG_OPTIONS.postfix=tls pgsql
PKG_OPTIONS.dovecot=kqueue pgsql ssl
DSPAM_STORAGE_DRIVER=sqlite
DSPAM_BINMODE=500
DSPAM_USER=vmail
DSPAM_GROUP=vmail
DSPAM_WWWUSER=vmail
DSPAM_WWWGROUP=vmail
```

Before starting the installation, create user *vmail* and group *vmail*, give both the id *5000* (choose whatever is best for you here). Then install all packages:

```
# cd /usr/pkgsrc/databases/postgresql84 && bmake install clean
# cd /usr/pkgsrc/mail/postfix && bmake install clean
# cd /usr/pkgsrc/mail/dovecot && bmake install clean
# cd /usr/pkgsrc/mail/wip/dovecot-antispam && bmake install clean
# cd /usr/pkgsrc/mail/dspam && bmake install clean
# cd /usr/pkgsrc/mail/postgrey && bmake install clean
```

We have to copy the startup scripts into */etc/rc.d/* as they are not installed by default:

```
# cp /usr/pkg/share/examples/rc.d/{pgsql,postfix,dovecot,postgrey} /etc/
    rc.d/
```

Enable all services in the */etc/rc.conf* file, so after a system reboot they will be started automatically, and to be able to start/stop them using the *rc.d* scripts.

```
sendmail_enable=NONE
pgsql=YES
postfix_enable=YES
dovecot_enable=YES
postgrey_enable=YES
```

3 Configuring the Database

At first init the database for using UTF8 as a template, so all newly created databases will be in UTF8 encoding, then startup the database.

```
# sh -c 'flags="-E UTF8" /etc/rc.d/pgsql initdb'
# /etc/rc.d/pgsql start
```

Setup a postgres user *maildb* and a database *maildb*. To do that, login as system user pgsql.

```
# su -l pgsql
pgsql> createuser -P -E maildb
(enter password further refered to as MAILDB_PASSWORD; answer all other
    questions with no)
pgsql> createdb -O maildb -E UTF8 maildb
```

Also as user pgsql secure database access by editing *pg_hba.conf*, comment all default settings:

```
# "local" is for Unix domain socket connections only
#local   all        all                            trust
# IPv4 local connections:
#host    all        all        127.0.0.1/32        trust
# IPv6 local connections:
#host    all        all        ::1/128             trust
```

and add this line:

```
host    maildb      maildb     127.0.0.1/32        md5
```

This will only allow access to the *maildb* database from localhost as user maildb by specifying the *MAILDB_PASSWORD* password. You don't need to do this step if you run a sealed mail server with no other services users. But if you allow other users to log into that box, this is neccessary!

We need to restart Postgres so that these changes take effect. Again as root:

```
# /etc/rc.d/pgsql restart
```

Now lets create the database schema in file schema.sql:

```
-- Which domains do we serve?
CREATE TABLE virtual_domains (
  id serial primary key,
  name varchar(50) not null unique
);

-- Which users do we have? A user corresponds to a mailbox.
CREATE TABLE virtual_users (
  id serial primary key,
  domain_id integer not null references virtual_domains(id)
    on delete cascade,
  password varchar(32) not null, -- md5
  email varchar(100) not null unique -- mailbox name
);

-- Aliases
CREATE TABLE virtual_aliases (
  id serial primary key,
  domain_id integer not null references virtual_domains(id)
    on delete cascade,
  source varchar(100) not null,
  destination varchar(100) not null
);
```

Load this into the *maildb* database with *psql -U maildb -f schema.sql maildb.*

4 Configuring Dovecot

We configure Dovecot before configuring Postfix, as Postfix uses Dovecot's SASL implementation and as such this should be set up before. In /usr/pkg/etc/dovecot create two files dovecot.conf and dovecot-sql.conf as seen below.

```
#
# dovecot.conf
#
protocols = imaps

protocol imap {
  ssl_listen = kvmdragon.ntecs.de:993
  mail_plugins = autocreate antispam
}

ssl_cert_file = /etc/ssl/server.pem
ssl_key_file = /etc/ssl/server.pem

mail_location = maildir:/mail/data/%d/%n

mail_uid = vmail
mail_gid = vmail

protocol lda {
  # Address to use when sending rejection mails (e.g. postmaster@example.com).
  postmaster_address = postmaster@ntecs.de

  # Hostname to use in various parts of sent mails, eg. in Message-Id.
  # Default is the system's real hostname.
  hostname = kvmdragon.ntecs.de

  # UNIX socket path to master authentication server to find users.
  auth_socket_path = /var/run/dovecot/auth-master
}

auth default {
  mechanisms = plain

  passdb sql {
    args = /usr/pkg/etc/dovecot-sql.conf
  }

  # XXX: Change to unpriviledged user that only makes Postgres connections!
  user = root

  socket listen {
    master {
      # Master socket provides access to userdb information. It's typically
      # used to give Dovecot's local delivery agent access to userdb so it
      # can find mailbox locations.
      path = /var/run/dovecot/auth-master
      mode = 0600
      # Default user/group is the one who started dovecot-auth (root)
      user = vmail
      group = vmail
    }
    client {
      path = /var/spool/postfix/private/auth
      mode = 0660
      user = postfix
      group = postfix
    }
  }
}

plugin {
  autocreate = Trash
  autocreate2 = Sent
  autocreate3 = Spam
  autosubscribe = Trash
  autosubscribe2 = Sent
  autosubscribe3 = Spam

  antispam_spam = Spam
  antispam_trash = Trash
  antispam_signature_missing = move
  antispam_dspam_binary = /usr/pkg/bin/dspam
  antispam_dspam_args = --user;%u
}

#
# dovecot-sql.conf
#
driver = pgsql
connect = host=127.0.0.1 dbname=maildb user=maildb password=
    MAILDB_PASSWORD
default_pass_scheme = PLAIN-MD5
password_query = SELECT email as user, password FROM virtual_users WHERE
    email='%u';
```

In *dovecot.conf* you have to change *ssl_ listen, postmaster_ address, hostname* to match your hostname and maybe the path given in *antispam_ dspam_ binary.* Also note that /etc/ssl/server.pem should be an existing self-signed password-less SSL certificate. The emails will be stored in subdirectories of */mail/data*, so this directory needs to be created and chown'ed to vmail:vmail. In dovecot-sql.conf change MAILDB_PASSWORD accrodingly. Give both files the correct permissions (dovecot.conf needs broader permissions as it is read by the local delivery agent; dovecot-sql.conf is only read by the dovecot process itself and should be root readable only):

```
# chmod 644 dovecot.conf
# chmod 600 dovecot-sql.conf
```

5 Configuring Postfix

Add the following lines at the end of Postfix main.cf:

```
#
# main.cf
#
myorigin = $mydomain
inet_interfaces = $myhostname, localhost
mydestination = $myhostname, localhost.$mydomain, localhost

virtual_mailbox_domains = pgsql:/usr/pkg/etc/postfix/pgsql-virtual-
    mailbox-domains.cf
virtual_mailbox_maps = pgsql:/usr/pkg/etc/postfix/pgsql-virtual-mailbox-
    maps.cf
virtual_alias_maps = pgsql:/usr/pkg/etc/postfix/pgsql-virtual-alias-maps.
    cf
```

```
virtual_transport=dspam
dspam_destination_recipient_limit=1

smtpd_use_tls=yes
# only allow secure auth
smtpd_tls_auth_only=yes
smtpd_tls_key_file = /etc/ssl/server.pem
smtpd_tls_cert_file = $smtpd_tls_key_file
smtpd_tls_CAfile = $smtpd_tls_key_file
smtpd_tls_loglevel = 0

#broken_sasl_auth_clients = yes
smtpd_sasl_auth_enable=yes
smtpd_sasl_type=dovecot
smtpd_sasl_path=private/auth
smtpd_recipient_restrictions=
  permit_mynetworks
  permit_sasl_authenticated
  reject_unknown_sender_domain
  reject_unauth_pipelining
  reject_unknown_recipient_domain
  reject_non_fqdn_sender
  reject_non_fqdn_recipient
  reject_non_fqdn_hostname
  reject_unauth_destination
  reject_unlisted_recipient
  check_policy_service inet:127.0.0.1:2525
```

Take care that no domain from the *virtual_ domains* is listed under the *mydestination* setting in *main.cf.* Otherwise the local delivery agent will be used instead of the virtual LDA we use!

Then in /usr/pkg/etc/postfix create the following three files (if you use a different path take care to change the filename in main.cf).

```
/usr/pkg/etc/postfix# cat pgsql-virtual-mailbox-domains.cf
user = maildb
password = MAILDB_PASSWORD
hosts = 127.0.0.1
dbname = maildb
query = SELECT 1 FROM virtual_domains WHERE name='%s'

/usr/pkg/etc/postfix# cat pgsql-virtual-mailbox-maps.cf
user = maildb
password = MAILDB_PASSWORD
hosts = 127.0.0.1
dbname = maildb
query = SELECT 1 FROM virtual_users WHERE email='%s'

/usr/pkg/etc/postfix# cat pgsql-virtual-alias-maps.cf
user = maildb
password = MAILDB_PASSWORD
hosts = 127.0.0.1
dbname = maildb
query = SELECT destination FROM virtual_aliases WHERE source='%s'
```

This tells Postfix how to get the desired information out of the Postgres database. The first defines the *virtual mailbox domains*, i.e. all domains that Postfix should accept mail for. If mail for other domains arrive they will be rejected.

The second is the *virtual mailbox maps* setting. Postfix needs to know, after resolving aliases, what are valid email addresses for the final destination. That is, we need one entry for each mailbox (which corresponds to a user).

The third is the *virtual alias maps* setting. For example if you want mneumann@ntecs.de and m.neumann@ntecs.de to be sent to the same mailbox mneumann@ntecs.de, you'd need the entry (source=m.neumann@ntecs.de, destination=mneumann@ntecs.de) in the *virtual_aliases* table. The purpose of the domain_id column in the table is only to guarantee a bit of referential integrity, other than that it has no meaning. Take care if you use *@catchall* arguments in the source column. Better not use it!

We have to secure those files as they contain passwords and only root should be able to read them.

```
# chmod 600 /usr/pkg/etc/postfix/*.cf
```

In *master.cf* we need the following two entries:

```
smtps     inet  n       -       n       -       -       smtpd
      -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes

dspam   unix  -       n       n       -       -       pipe
    flags=DRhu user=vmail:vmail argv=/usr/pkg/bin/dspam --user ${
        recipient} --deliver=innocent -f ${sender} -d ${recipient}
```

Take care about line breaks as they are wrong in this document!

The first adds TLS-encrypted SMTP. This can come in handy in networks where port 25 is blocked, like my university network! Even worse, they don't block the port totally but honeypot it, so if you try to connect to your SMTP server you get connected to their server and get trapped! SMTPS is listening on a different port and is usually not blocked so you can still send email. That port is not used to receive email, so it's a perfect choice!

The second line (dspam) adds a local delivery agent (LDA) that we use to deliver mail. Every email gets passed to the dspam binary which will forward it to the dovecot local delivery agent as we will see for final delivery into the users mailbox. We only deliver innocent emails, other emails (SPAM) will be quarantined, which in our case means, sent to the Spam folder.

You might have to change the *check_policy_service* port in *main.cf* depending on the port postgrey listens! If you don't and postgrey uses a different port, all mail will be rejected! Take care that postgrey is always running! That's all we have to do for postfix.

6 Dspam

We will use Dspam for spam detection. We used the sqlite driver for dspam to make dspam_clean work properly. Basically any SQL-based driver will work, the default filesystem driver won't. Once again take a look at the *Installation* section and see those DSPAM_* flags used during compilation. It's *important* that dspam gets installed as user vmail and internally knows that it's using the vmail user instead of the default dspam user. We could use groups to avoid that, but this way is much easier. Postfix will execute the dspam binary as user vmail and dspam will then, after doing spam analysis, execute the /usr/pkg/libexec/dovecot/delivery LDA binary, which will read dovecot.conf (that's why it must be readable by vmail!) and use the mail_location setting to determine where to store the mail.

Edit /usr/pkg/etc/dspam/dspam.conf:

```
#
# The important settings:
#
Home /var/dspam
TrustedDeliveryAgent "/usr/pkg/libexec/dovecot/deliver"
QuarantineAgent "/usr/pkg/libexec/dovecot/deliver -d %u -m Spam"
OnFail unlearn
Trust root
Trust vmail
Preference "spamAction=quarantine"
Preference "signatureLocation=headers"
LocalMX kvmdragon.ntecs.de

#
# Custom settings
#
Notifications   off
TrainingMode teft
TestConditionalTraining on
Feature whitelist
Algorithm graham burton
Tokenizer chain
PValue bcr
PurgeSignatures 14          # Stale signatures
PurgeNeutral    90          # Tokens with neutralish probabilities
PurgeUnused     90          # Unused tokens
PurgeHapaxes    30          # Tokens with less than 5 hits (hapaxes)
PurgeHits1S     15           # Tokens with only 1 spam hit
PurgeHits1I     15           # Tokens with only 1 innocent hit
SystemLog off
UserLog   off
Opt out
MaxMessageSize 4194304
ProcessorURLContext on
ProcessorBias on
```

You might need to change the *LocalMX* setting to the IP the postfix server is using. The *TrustedDeliveryAgent* is used for innocent (i.e. non-spam) emails. It will be called with the flags that are given in Postfix's master.cf entry for dspam (-f ${sender} -d ${recipient}). The *QuarantineAgent* is what is called for spam emails. Here the flags from Postfix are not passed, that is why we need to specify "-d %u" (deliver to the corresponding user). The "-m Spam" tells Dovecot's LDA to deliver spam into the Spam folder.

Finally make sure that the *dspam.conf* file is readable by *vmail:vmail.*

```
# chown vmail:vmail /usr/pkg/etc/dspam/dspam.conf
```

As the administrator I want to get weekly statistics how the spam filter performed in terms of false positives etc. The script *dspam_stats* returns those statistics. We also want to clean up the dspam databases every week or so. For that, we set up the file */etc/weekly.local* which gets called automatically by *periodic* every week:

```
#!/bin/sh
echo
echo "Dspam statistics:"
echo
/usr/pkg/bin/dspam_stats -H

echo
echo "Dspam clean"
/usr/pkg/bin/dspam_clean -s -p -u
```

We make that file executable. We'll get the output of this script automatically sent to the mailbox of root (for whom we should have an alias for in the virtual_aliases table to a valid mailbox!).

7 Postgrey

Postgrey is a greylisting policy agent for Postfix. It helps to filter out a lot of spam *before* it enters the system. There are numerous other policy agents for Postfix available, but this one seems to be the easiest to install. Note that if this service goes down no mail will be accepted anymore, so it is critical to the whole mail system!

There is nothing to configure.

8 Start all services

You already have all services enabled in */etc/rc.conf*, so it's easy to start them.

```
# /etc/rc.d/dovecot start
# /etc/rc.d/postgrey start
# /etc/rc.d/postfix start
## /etc/rc.d/pgsql start # database already started!
```

9 Adding a domain/user

Connect as user maildb to database maildb (psql -h 127.0.0.1 -U maildb maildb). We want to create a new mailbox info@mydomain.com and an alias for webmaster@mydomain.com.

```
# At first create a new domain
insert into virtual_domains (name) values ('mydomain.com');

# Then add a user/mailbox
insert into virtual_users (domain_id, password, email) values (
  select id from virtual_domains where name = 'mydomain.com',
  md5(’password’),
  ’info@mydomain.com’);

# From now on mail to info@mydomain.com would arrive in the mailbox.
# We want an additional alias.
insert into virtual_aliases (domain_id, source, destination) values (
  select id from virtual_domains where name = ’mydomain.com’,
  ’webmaster@mydomain.com’,
  ’info@mydomain.com’);

# You could add further aliases here.
```

10 Todo

- Explain how to generate SSL PEM file /etc/ssl/server.pem.
- SPF records
- dspam: clamav integration
- dovecot: purge SPAM/Trash regularily
- Explain the dovecot antispam plugin.
- Explain MX records, reverse DNS entries etc.