Creating a Virtual Mail Server with Postfix, Dovecot and MySQL

This article shows how to build and set up a secure virtual mail server using Postfix, Dovecot and MySQL on a Slackware 14.1 platform.

Because the build and installed platforms may very likely be different, and the configuration can seem complex to those not familiar with postfix, dovecot and mysql, I have organized this article in a way that I hope will help you separate and test individual tasks, without losing your way!

The article has been written as the result of my own experience in setting up a virtual mail server on my own Linode.com VPS, but I have avoided many overt references to Linode.com specific tools or configurations, so it should be easily adapted for any platform. The only Linode.com reference will be to list a few additional packages that will be needed for their Slackware 14.1 base.

Our Target Virtual Mail Server

What we mean by *virtual mail server* is that mail boxes need not correspond to user accounts on the machine, but instead belong to virtual users defined in the database. This allows for serving mail for many mail boxes and across multiple domains.

Running a virtual mail server may not be the best choice for many use cases as it presents some additional security risks and requirements! In particular, you should not run a virtual mail server on a machine which provides login accounts for multiple users! A virtual mail MTA should only be configured on a well secured host dedicated to the purpose!

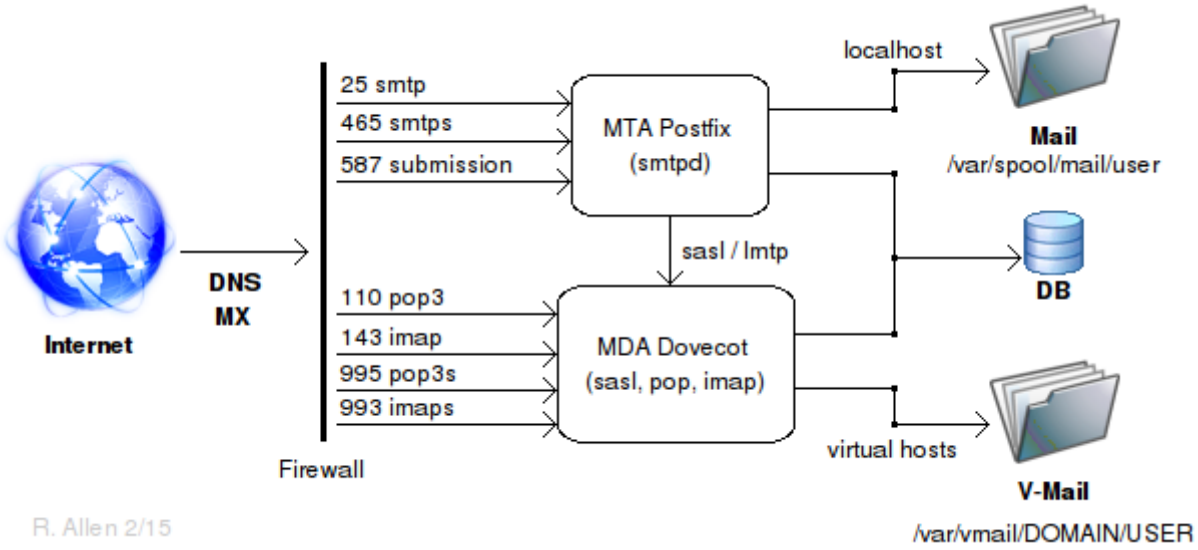
For security we will configure Dovecot to provide authentication for Postfix via SASL based on virtual mail box credentials. We will also require TLS/SSL for all user access, and of course will assure that it does not turn into a spam serving open relay!

To get started you simply need a Slackware 14.1 base installation, with updated security patches, particularly for bash, openssl, openssl-solibs and gnutls.

If you are installing to a Linode.com VPS with their Slackware 14.1 base, you will need to add a few missing packages in addition to mariadb, dovecot and postfix. During my own set up (which includes fail2ban, not covered here) I needed to add: db48, cyrus-sasl, libaio, pidentd and sqlite

There are many parts to our virtual mail server which will be covered in separate sections. But the best way to organize them all is with a simple graphic showing the major installed components and their relationships. Figure 1, below, represents an overall view of the target virtual mail system described by this article.

Figure 1: Virtual Email Overview



Component Installation

Listing of major items from Figure 1 provides a todo check-list for installing the major components of our virtual mail server.

Follow the links listed here and *carefully complete each item*. When finished you will have a secure, working virtual mail server which may then be further configured to meet your specific needs!

- Firewall configuration
- Database initialization
- Required users, groups and paths
- SSL configuration and certificates
- Postfix configuration
- Dovecot configuration
- DNS, MX records and hosts file
- Startup and Troubleshooting

DNS, MX Records and Hosts File

In order for your mail server to be found on a wider network, you must correctly configure the DNS and MX records for your host's domain. These records are external to the host itself and so are outside the scope of this article.

In general, you will need to configure the *DNS zone file* for your domain and create MX records within that zone file which point to your mail server. If your target system is a hosted platform such as a VPS, then the provider will supply the tools and guides for doing so. You must also be sure to provide a PTR record for reverse address lookup, many systems will not receive mail from hosts which they cannot identify.

An internet search will also produce many helpful resources, which I will leave to you as an exercise.

In addition, on the host machine itself, you will need to be sure that the hosts file includes the corresponding lines for your IP address, hostname, FQDN and domain as they are used by postfix and dovecot for resolving the disposition of messages:

```
cat /etc/hosts
...
127.0.0.1       localhost
127.0.0.1       localhost.localdomain   localhost
xx.xx.xx.xx     darkstar
xx.xx.xx.xx     darkstar.my-domain.com   my-domain.com
```

Startup and Troubleshooting

After you have completed each of the major tasks listed above from Figure 1, you should be ready to startup your virtual mail server.

You should verify each aspect of the system separately and in order, so that you can resolve any potential problems methodically.

If something seems to not work as expected, *take the time to understand why* and avoid the urge to just “try this or that” - you will only corrupt and compromise your system by that method!

First, lets start dovecot and postfix and look for any error messages - there really should be none.

```
chmod +x /etc/rc.d/rc.dovecot
/etc/rc.d/rc.dovecot start
Firing up dovecot

tail /var/log/maillog
...
Feb 16 01:48:45 darkstar dovecot: master: Dovecot v2.2.13 starting up for
imap, pop3, lmtp (core dumps disabled)
```

If you receive any other messages or errors from the shell or the log file, investigate and fix them before continuing!

Next let's do the same for postfix:

```
chmod +x /etc/rc.d/rc.postfix
/etc/rc.d/rc.postfix start
postfix/postfix-script: starting the Postfix mail system

tail /var/log/maillog
...
Feb 16 01:51:55 darkstar postfix/postfix-script[6931]: starting the Postfix
mail system
Feb 16 01:51:55 darkstar postfix/master[6933]: daemon started -- version
```

```
2.11.3, configuration /etc/postfix
```

Same as before, if you receive any other messages or errors from the shell or the log file, investigate and fix them before continuing!

Next, with postfix and dovecot running, use netstat to confirm that things are as expected:

```
netstat -plntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
...
tcp        0      0 0.0.0.0:587             0.0.0.0:*               LISTEN
6933/master
tcp        0      0 0.0.0.0:465             0.0.0.0:*               LISTEN
6933/master
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
6933/master
tcp        0      0 0.0.0.0:993             0.0.0.0:*               LISTEN
6843/dovecot
tcp        0      0 0.0.0.0:995             0.0.0.0:*               LISTEN
6843/dovecot
```

You should see master (the postfix master process) listening on ports 25, 465 and 587, and dovecot on ports 993 and 995.

You should also verify that your firewall rules are in fact loaded - you do not want to run your mail server without the firewall, even briefly!

```
iptables -L
...
```

If all looks good so far, check whether you can connect to the mailserver from the outside world. We should use port 25 for this, but many ISPs block port 25 so it is difficult to get a reliable test result. So see if you can connect to port 465 or 587 using telnet.

Type the line “telnet my-domain.com 465” and hit enter key:

```
telnet my-domain.com 465
Trying xx.xx.xx.xx...
Connected to my-domain.com.
Escape character is '^]'.
220 darkstar.my-domain.com ESMTP Postfix

# Use Ctl-] quit to end the connection
```

If you do not receive a response similar to that shown then it is likely that your host is not reachable from your location.

Most commonly this will be due to a bad DNS record or incorrect nameserver settings at the domain

registrar. Verify that these are correct.

It can also result from incorrect or conflicting firewall rules. Remember that iptables rules are evaluated from top to bottom and the first match determines the fate of each request. Look carefully to see if any pre-existing rules may be dropping your requests before they reach the intended rule!

If it connects then you are ready to set up an inbox in a mail client like Thunderbird and give it a final test!

When you create an inbox in your mail client, remember to use the secure imap and pop3 connections - we are not listening on the insecure service ports. You must use the full mailbox name, name@my-domain.com, as the user name, and the password must be that used in the MySQL INSERT or UPDATE queries which created the virtual user.

And remember - although this configuration is as secure as the machine it is running on, it is still a minimal configuration!

You will also definitely want to install fail2ban (available from SBo). You will also want a spam filtering application and possibly even a virus detection program for incoming and outgoing email.

You should think carefully about your own specific requirements, read the postfix and dovecot documentation thoroughly, subscribe to their mail lists and learn to interpret the logs. Watch the system carefully and take it offline at the first sign of compromise!

Running an email server on the internet is definitely not a click and forget past time!

Sources

- Originally written by astrogeek

howtos, email, postfix, dovecot, mysql, ssl

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:network_services:postfix_dovecot_mysql

Last update: **2015/08/29 09:45 (UTC)**