

Virtualize user and domain accounts

In the following article, TipsMake.com will guide, explain to you how to install Postfix-based mail server system - built on virtual user and domain accounts

In the following article, TipsMake.com will guide, explain to you how to install Postfix-based mail server system - built on virtual user and domain accounts (eg these 2 values ??are stored in the server) MySQL database). At the same time, we also rebuilt the whole process of installing and setting up Courier (Courier-POP3, Courier-IMAP), through which Courier can identify the same information flow as MySQL.

The result of Postfix server is the ability to support authentication mechanisms SMTP-AUTH, TLS and quota (quota is not integrated with Postfix in default mode), where the password will be stored under the encrypted form in the engine. database. In addition, the article will show how to install Amavisd, SpamAssassin and ClamAV to scan emails and detect associated threats.

Preliminary note

The system used here is based on the x86_64 Fedora 13 server platform, has a static IP address 192.168.0.100 and hostname is *server1.example.com* . In addition, turn off the firewall and SELinux feature.

Install some required software

First, we need to update a number of application packages on the system:

```
yum update
```

```
yum groupinstall 'Development Tools'
```

```
yum groupinstall 'Development Libraries'
```

Install Apache, MySQL, phpMyAdmin

With only one command line, there are packages required to build the Courier-IMAP system later:

```
yum install ntp httpd mysql-server php php-mysql php-mbstring rpm gcc build mysql-devel openssl-devel cyrus-s-devel pkgconfig zlib-devel phpMyAdmin pcre-devel openldap-devel postgresql-devel expect libtool-ltdl-devel openldap -servers libtool gdbm-devel pam-devel gamin-devel
```

Courier-IMAP, Courier-Authlib, and Maildrop installation

It is inconvenient when there is no rpm package for Courier-IMAP, Courier-Authlib, and Maildrop, so we need to build the source code ourselves. Note that RPM packages should not be created with root accounts, courier-

imap will automatically turn off the compilation process if it detects that the compiler is running with the root account. Therefore, we need to create a normal account (in this example, falko) and create an accompanying password:

```
useradd -m -s /bin /bash falko  
passwd falko
```

We will need sudo syntax later, so that falko account can compile and install rpm packages. But first, we must allow the falko account to run all commands using sudo:

```
visudo
```

In the file just opened, there will be *root ALL = (ALL) ALL* . Add a similar line for falko directly below:

```
[.]  
## Allow root to run any commands anywhere  
root ALL = (ALL) ALL  
falko ALL = (ALL) ALL  
[.]
```

And now, we're ready to build rpm packages. First, log in to the falko account:

```
su falko
```

Create environment variables:

```
mkdir $HOME / rpm  
mkdir $HOME / rpm / SOURCES  
mkdir $HOME / rpm / SPECS  
mkdir $HOME / rpm / BUILD  
mkdir $HOME / rpm / BUILDROOT  
mkdir $HOME / rpm / SRPMS  
mkdir $HOME / rpm / RPMS  
mkdir $HOME / rpm / RPMS / i386  
mkdir $HOME / rpm / RPMS / x86_64  
  
echo "%_topdir $HOME / rpm" >> $HOME / .rpmmacros
```

Create the downloads folder and download the source code from here:

```
mkdir $HOME / downloads  
cd $HOME / downloads  
  
wget https://sourceforge.net/projects/courier/files/authlib/0.63.0/courier-authlib-0.63.0.tar.bz2/download  
wget https://sourceforge.net/projects/courier/files/imap/4.8.0/courier-imap-4.8.0.tar.bz2/download
```

```
wget https://sourceforge.net/projects/courier/files/maildrop/2.5.0/maildrop-2.5.0.tar.bz2/download
```

Still in the \$ HOME / downloads folder, build courier-authlib package:

```
sudo rpmbuild -ta courier-authlib-0.63.0.tar.bz2
```

Then, the rpm package will be in \$ HOME / rpm / RPMS / x86_64 (\$ HOME / rpm / RPMS / i386 if you use i386):

```
cd $ HOME / rpm / RPMS / x86_64
```

Use the command:

```
ls -l
```

to list the appropriate rpm packages:

```
[falko @ server1 x86_64] $ ls -l
total 516
-rw-r - r-- 1 root root 124304 Jun 10 17:48 courier-authlib-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 258896 Jun 10 17:48 courier-authlib-debuginfo-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 35064 Jun 10 17:48 courier-authlib-devel-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 17424 Jun 10 17:48 courier-authlib-ldap-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13956 Jun 10 17:48 courier-authlib-mysql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13120 Jun 10 17:48 courier-authlib-pgsql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 8328 Jun 10 17:48 courier-authlib-pipe-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 34160 Jun 10 17:48 courier-authlib-userdb-0.63.0-1.fc13.x86_64.rpm
[falko @ server1 x86_64] $
```

Select an appropriate package and install the following:

```
sudo rpm -ivh courier-authlib-0.63.0-1.fc13.x86_64.rpm courier-authlib-mysql-0.63.0-1.fc13.x86_64.rpm
courier-authlib-devel-0.63.0-1.fc13.x86_64.rpm
```

Next, go back to the downloads folder:

```
cd $ HOME / downloads
```

and run rpmbuild again, but there is no sudo otherwise the compilation will fail:

```
rpmbuild -ta courier-imap-4.8.0.tar.bz2
```

After this process, the rpm package will be found in the HOME / rpm / RPMS / x86_64 folder (\$ HOME / rpm / RPMS / i386 if you are using the i386 system):

```
cd $HOME / rpm / RPMS / x86_64
```

Use the command:

```
ls -l
```

will list the appropriate rpm packages:

```
[falko @ server1 x86_64] $ ls -l
total 2300
-rw-r - r-- 1 root root 124304 Jun 10 17:48 courier-authlib-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 258896 Jun 10 17:48 courier-authlib-debuginfo-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 35064 Jun 10 17:48 courier-authlib-devel-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 17424 Jun 10 17:48 courier-authlib-ldap-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13956 Jun 10 17:48 courier-authlib-mysql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13120 Jun 10 17:48 courier-authlib-pgsql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 8328 Jun 10 17:48 courier-authlib-pipe-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 34160 Jun 10 17:48 courier-authlib-userdb-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 falko falko 632296 Jun 10 17:58 courier-imap-4.8.0-1.13.x86_64.rpm
-rw-r - r-- 1 falko falko 1188976 Jun 10 17:58 courier-imap-debuginfo-4.8.0-1.13.x86_64.rpm
[falko @ server1 x86_64] $
```

And courier-imap installation as follows:

```
sudo rpm -ivh courier-imap-4.8.0-1.13.x86_64.rpm
```

Go back to downloads folder:

```
cd $HOME / downloads
```

continue running rpmbuild, this time to build maildrop package:

```
sudo rpmbuild -ta maildrop-2.5.0.tar.bz2
```

When completed, the rpm package will be in the \$HOME / rpm / RPMS / x86_64 folder (\$HOME / rpm / RPMS / i386 if you use the i386 system):

```
cd $HOME / rpm / RPMS / x86_64
```

Type the command:

```
ls -l
```

to list the appropriate packages:

```

[falko @ server1 x86_64] $ ls -l
total 5400
-rw-r - r-- 1 root root 124304 Jun 10 17:48 courier-authlib-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 258896 Jun 10 17:48 courier-authlib-debuginfo-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 35064 Jun 10 17:48 courier-authlib-devel-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 17424 Jun 10 17:48 courier-authlib-ldap-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13956 Jun 10 17:48 courier-authlib-mysql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 13120 Jun 10 17:48 courier-authlib-pgsql-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 8328 Jun 10 17:48 courier-authlib-pipe-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 34160 Jun 10 17:48 courier-authlib-userdb-0.63.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 falko falko 632296 Jun 10 17:58 courier-imap-4.8.0-1.13.x86_64.rpm
-rw-r - r-- 1 falko falko 1188976 Jun 10 17:58 courier-imap-debuginfo-4.8.0-1.13.x86_64.rpm
-rw-r - r-- 1 root root 1759056 Jun 10 18:06 maildrop-2.5.0-1.13.x86_64.rpm
-rw-r - r-- 1 root root 1243400 Jun 10 18:06 maildrop-debuginfo-2.5.0-1.13.x86_64.rpm
-rw-r - r-- 1 root root 99764 Jun 10 18:06 maildrop-devel-2.5.0-1.13.x86_64.rpm
-rw-r - r-- 1 root root 62536 Jun 10 18:06 maildrop-man-2.5.0-1.13.x86_64.rpm
[falko @ server1 x86_64] $

```

And set up maildrop as follows:

```
sudo rpm -ivh maildrop-2.5.0-1.13.x86_64.rpm
```

After completing the above processes, log into the system with the root account:

```
exit
```

Apply Quote patch with Postfix

Here, we will have to find a way to patch Quota with Postfix in the following way: download the source code, apply the patch, build the rpm installation package of Postfix and install:

```

cd /usr/src
wget http://ftp-stud.fht-esslingen.de/pub/Mirrors/fedora/linux/releases/13/Fedora/source/SRPMS/postfix-2.7.0-1.fc13.src.rpm
rpm -ivh postfix-2.7.0-1.fc13.src.rpm

```

The last statement will show a few warnings, you can ignore these warnings:

```

warning: mockbuild user does not exist - dùng root
c?nh báo: mockbuild group không t?n t?i - dùng root

```

```

cd /root/rpmbuild/SOURCES
wget http://vda.sourceforge.net/VDA/postfix-vda-2.7.0.patch
cd /root/rpmbuild/SPECS/

```

Next, edit the postfix.spec file:

en postfix.spec

Add Patch0 syntax: postfix-vda-2.7.0.patch to # Patches location, and *% patch0 -p1 -b .vda* to *% setup -q* :

```
[.]
Patches

Patch0: postfix-vda-2.7.0.patch
Patch1: postfix-2.7.0-config.patch
Patch2: postfix-2.6.1-files.patch
Patch3: postfix-alternatives.patch
Patch8: postfix-large-fs.patch
Patch9: pflogsumm-1.1.2-datecalc.patch
[.]
% prep
% setup -q
# Apply obligatory patches
% patch0 -p1 -b .vda
% patch1 -p1 -b .config
% patch2 -p1 -b .files
% patch3 -p1 -b .alternatives
% patch8 -p1 -b .large-fs
[.]
```

Then, build the rpm installation package of Postfix with quota and MySQL support:

```
rpmbuild -ba postfix.spec
```

The Postfix installation rpm package is created in */ root / rpmbuild / RPMS / x86_64* (*/ root / rpmbuild / RPMS / i386* if you use the i386 system):

```
cd / root / rpmbuild / RPMS / x86_64
```

Use the command:

```
ls -l
```

to list the appropriate packages:

```
[root @ server1 x86_64] # ls -l
total 8804
-rw-r - r-- 1 root root 2197708 Jun 10 18:17 postfix-2.7.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 6746304 Jun 10 18:17 postfix-debuginfo-2.7.0-1.fc13.x86_64.rpm
-rw-r - r-- 1 root root 61460 Jun 10 18:17 postfix-perl-scripts-2.7.0-1.fc13.x86_64.rpm
[root @ server1 x86_64] #
```

Select the appropriate Postfix package and install it using the following command:

```
rpm -ivh postfix-2.7.0-1.fc13.x86_64.rpm
```

Set password for MySQL and set up phpMyAdmin

Start MySQL:

```
chkconfig --levels 235 mysqld on  
/etc/init.d/mysqld start
```

Set password for MySQL root account:

```
mysql_secure_installation
```

```
[root @ server1 ~] # mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In hàng l?nh ?? ??ng nh?p vào MySQL ?? b?o v? nó, s? c?n ph?i hi?n th?i password for the root user. If you've just installed MySQL, và b?n không ??t ???c m?t kh?u root này, m?t kh?u s? ???c tr?ng, so you should just press enter here.

*Enter current password cho root (nh?p cho không):
OK, ?ã ???c s? d?ng m?t kh?u và chuy?n ??i .*

Thi?t l?p m?t kh?u g?c c?n thi?t mà không th? ??ng nh?p vào MySQL root user without the proper authoring.

*Set root password? [Y/n]
New password: ? set root sql password
Re-enter new password: Password updated successfully!
Reloading privilege tables .
. Success!*

Vì m?c ??nh, m?t cài ??t MySQL có m?t ng??i dùng không rõ, Allow anyone ?? ??ng nh?p vào MySQL không có ?? ng??i dùng ?ã t?o cho ng??i dùng more. ?ây ???c s? d?ng ch? ?? ki?m tra, và ?? t?o cài ??t go a bit smoother. B?n nên g? b? chúng tr??c khi chuy?n vào m?t production environment.

Remove anonymous users? [Y/n]

. Success!

Th??ng, root nên ch? ???c phép phép k?t n?i t? 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login command? [Y/n]

. Success!

By default, MySQL comes with a database ???c xác ??nh 'th?' mà có th? access. ?ây là ch? ??nh ch? ch? cho th? th?, và nên ???c g? b? tr???c khi chuy?n vào m?t Production environment.

Remove test database and access to it? [Y/n]

- Dropping test database .

. Success!

- Removing privileges on test database .

. Success!

Reloading the privilege tables will ensure that all changes do so far s? x? lý ngay ngay.

Reload privilege tables now? [Y/n]

. Success!

Cleaning up .

All done! If you've completed all steps theo ?ây, b?n MySQL cài ??t nên ???c b?o v?.

Thanks for dùng MySQL!

[root @ server1 ~] #

Next, we have to set up phpMyAdmin, change Apache so that phpMyAdmin accepts connections not only from localhost:

en /etc/httpd/conf.d/phpMyAdmin.conf

phpMyAdmin - Web based MySQL browser written in php

#

Phép phép ch? có ??a ch? t? m?c ??nh

#

But allowing phpMyAdmin to anyone other localhost should be considered

dangerous unless properly secured by SSL

```
Alias ?/?/phpMyAdmin /usr/share/phpMyAdmin
Alias ?/?/phpmyadmin /usr/share/phpMyAdmin
#
# order deny, allow
# deny from all
# allow from 127.0.0.1
# allow from :: 1
#

# Th? m?c này không có quy?n truy c?p qua HTTP - ???c l?y t? g?c
# phpMyAdmin upstream tarball
#
```

```
Order Deny, Allow
Deny from All
Allow from None
```

```
# C?u hình c?u hình mod_security t?i phpMyAdmin directories t?
# filtering SQL etc. This may break your mod_security implementation.
#
#
#
# SecRuleInheritance Off
#
#
```

Create boot path for Apache and activate:

```
chkconfig --levels 235 httpd on
/etc/init.d/httpd start
```

Open the browser and type in `http://server1.example.com/phpMyAdmin/` or `http://192.168.0.100/phpMyAdmin/` , log in with the root account and the MySQL password declared above.

Create MySQL database for Postfix / Courier

Here, we will create the database with the name mail:

```
mysqladmin -u root -p create mail
```

Move to MySQL shell:

```
mysql -u root -p
```

And here, we will create mail_admin account with mail_admin_password password (replace with your optional password) with basic rights such as SELECT, INSERT, UPDATE, and DELETE on the mail database. This account will be used by Postfix and Courier to connect to the mail database:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON mail. * TO 'mail_admin' @ 'localhost' IDENTIFIED
BY 'mail_admin_password';
GRANT SELECT, INSERT, UPDATE, DELETE ON mail. * TO 'mail_admin'@'localhost.localdomain'
IDENTIFIED BY 'mail_admin_password';
FLUSH PRIVILEGES;
```

Create tables that Postfix and Courier need:

```
USE mail;
```

```
CREATE TABLE domains (
domain varchar (50) NOT NULL,
PRIMARY KEY (domain))
TYPE = MyISAM;
```

```
CREATE TABLE forwardings (
source varchar (80) NOT NULL,
destination TEXT NOT NULL,
PRIMARY KEY (source))
TYPE = MyISAM;
```

```
CREATE TABLE users (
email varchar (80) NOT NULL,
password varchar (20) NOT NULL,
bigint quota (20) DEFAULT '10485760',
PRIMARY KEY (email)
) TYPE = MyISAM;
```

```
CREATE TABLE transport (
domain varchar (128) NOT NULL default "",
transport varchar (128) NOT NULL default "",
UNIQUE KEY domain (domain)
) TYPE = MyISAM;
```

```
quit;
```

With the command quit; we will exit the MySQL shell and return to the Linux shell.

Domain data tables store each virtual domain that Postfix uses to receive email (eg *example.com*).

The forwardings table is for emails pointing to another email, for example pointing from *info@example.com* to *sales@example.com*

The users table stores all virtual account information and password with the mail box quota value (in this example, the default value is 10485760 bytes equivalent to 10MB).

Transport table is an additional option, for advanced users, to allow mail forwarding for each single user, or the entire domain as well as all mail to another server.

Adjust Postfix

Next, we have to show Postfix to find all the information in the database, so we have to create 6 text files. Postfix will connect to MySQL with the IP address: 127.0.0.1 inside localhost:

```
vi /etc/postfix/mysql-virtual_domains.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
query = SELECT domain AS virtual FROM domains WHERE domain = '% s'
hosts = 127.0.0.1
vi /etc/postfix/mysql-virtual_forwardings.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
câu h?i = SELECT ?ích ??n t? cu?i cùng WHERE ngu?n = '% s'
hosts = 127.0.0.1
vi /etc/postfix/mysql-virtual_mailboxes.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
query = SELECT CONCAT (SUBSTRING_INDEX (email, '@', - 1), '/', SUBSTRING_INDEX (email, '@',
1), '/') FROM users WHERE email = '% s'
hosts = 127.0.0.1
vi /etc/postfix/mysql-virtual_email2email.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
query = SELECT email FROM users WHERE email = '% s'
hosts = 127.0.0.1
en /etc/postfix/mysql-virtual_transports.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
query = SELECT transport FROM transport WHERE domain = '% s'
hosts = 127.0.0.1
vi /etc/postfix/mysql-virtual_mailbox_limit_maps.cf
```

```
user = mail_admin
password = mail_admin_password
dbname = mail
query = SELECT quota FROM WHERE email = '% s'
hosts = 127.0.0.1
chmod o = /etc/postfix/mysql-virtual_*.cf
chgrp postfix /etc/postfix/mysql-virtual_*.cf
```

Create a user account and a group called vmail with the / home / vmail root directory. This will be the place to store all mail boxes.

```
groupadd -g 5000 vmail
useradd -g vmail -u 5000 vmail -d / home / vmail -m
```

Next, we need to modify some parameters of Postfix, make sure that you have replaced the value server1.example.com with the corresponding FQDN, otherwise Postfix will not work properly:

```
postconf -e 'myhostname = server1.example.com'
postconf -e 'mydestination = server1.example.com, localhost, localhost.localdomain'
postconf -e 'mynetworks = 127.0.0.0/8'
postconf -e 'virtual_alias_domains ='
postconf -e 'virtual_alias_maps = proxy: mysql: /etc/postfix/mysql-virtual_forwardings.cf, mysql:
/etc/postfix/mysql-virtual_email2email.cf'
postconf -e 'virtual_mailbox_domains = proxy: mysql: /etc/postfix/mysql-virtual_domains.cf'
postconf -e 'virtual_mailbox_maps = proxy: mysql: /etc/postfix/mysql-virtual_mailboxes.cf'
postconf -e 'virtual_mailbox_base = / home / vmail'
postconf -e 'virtual_uid_maps = static: 5000'
postconf -e 'virtual_gid_maps = static: 5000'
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'broken_sasl_auth_clients = yes'
postconf -e 'smtpd_sasl_authenticated_header = yes'
postconf -e 'smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated,
reject_unauth_destination'
postconf -e 'smtpd_use_tls = yes'
postconf -e 'smtpd_tls_cert_file = /etc/postfix/smtpd.cert'
postconf -e 'smtpd_tls_key_file = /etc/postfix/smtpd.key'
postconf -e 'transport_maps = proxy: mysql: /etc/postfix/mysql-virtual_transports.cf'
postconf -e 'virtual_create_maildirsize = yes'
postconf -e 'virtual_maildir_extended = yes'
postconf -e 'virtual_mailbox_limit_maps = proxy: mysql: /etc/postfix/mysql-
virtual_mailbox_limit_maps.cf'
postconf -e 'virtual_mailbox_limit_override = yes'
postconf -e 'virtual_maildir_limit_message = "The user you are trying to reach is over quota."'
postconf -e 'virtual_overquota_bounce = yes'
postconf -e 'proxy_read_maps = $ local_recipient_maps $ virtual_alias_maps $ virtual_alias_domains $
virtual_mailbox_maps $ virtual_mailbox_domains $ relay_recipient_maps $ relay_domains $
canonical_maps $ sender_canonical_maps $ recipient_canonical_maps $ relocated_maps $
```

```
transport_maps $ mynetworks $ virtual_mailbox $ transportation_maps $ mynetworks $
virtual_mailbox_limit_maps'
postconf -e 'inet_interfaces = all'
```

Then create the necessary SSL authentication mechanism for TLS:

```
cd / etc / postfix
openssl req -new -outform PEM -out smtpd.cert -newkey rsa: 2048 -nodes -keyout smtpd.key -keyform
PEM -days 365 -x509
```

```
Country Name (2 letter code) [XX]:
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Tên tên (eg, c?a tên c?a b?n, máy ch? hostname) []:
Email Address []:
```

Then change the limits of smtpd.key:

```
chmod o = /etc/postfix/smtpd.key
```

Adjustment Saslauthd

Change `/usr/lib64/sasl2/smtpd.conf` (`/usr/lib/sasl2/smtpd.conf` if you use the i386 system), it will look like this:

```
vi /usr/lib64/sasl2/smtpd.conf

pwcheck_method: authdaemond
log_level: 3
mech_list: PLAIN LOGIN
authdaemond_path: / var / spool / authdaemond / socket
```

Turn off the Sendmail function and start Postfix, saslauthd, and courier-authlib:

```
chmod 755 / var / spool / authdaemond
chkconfig --levels 235 courier-authlib on
/etc/init.d/courier-authlib start
```

```
chkconfig --levels 235 sendmail off
chkconfig --levels 235 postfix on
chkconfig --levels 235 saslauthd on
/etc/init.d/sendmail stop
/etc/init.d/postfix start
/etc/init.d/saslauthd start
```

Adjust Courier

Next, we need to specify Courier to authenticate identity information from the MySQL database. First, edit the file `/etc/authlib/authdaemonrc` and replace the `authmodulelist` value:

```
en / etc / authlib / authdaemonrc
```

```
[.]  
authmodulelist = "authmysql"  
# authmodulelist = "authuserdb authpam authpgsql authldap authmysql authcustom authpipe"  
[.]
```

Continue to edit the file `/etc/authlib/authmysqlrc` :

```
cp / etc / authlib / authmysqlrc / etc / authlib / authmysqlrc_orig  
cat / dev / null > / etc / authlib / authmysqlrc  
en / etc / authlib / authmysqlrc
```

```
MYSQL_SERVER localhost  
MYSQL_USERNAME mail_admin  
MYSQL_PASSWORD mail_admin_password  
MYSQL_PORT 0  
MYSQL_DATABASE mail  
MYSQL_USER_TABLE users  
MYSQL_CRYPT_PWFIELD password  
#MYSQL_CLEAR_PWFIELD password  
MYSQL_UID_FIELD 5000  
MYSQL_GID_FIELD 5000  
MYSQL_LOGIN_FIELD email  
MYSQL_HOME_FIELD "/ home / vmail"  
MYSQL_MAILDIR_FIELD CONCAT (SUBSTRING_INDEX (email, '@', - 1), '/', SUBSTRING_INDEX  
(email, '@', 1), '/')  
#MYSQL_NAME_FIELD  
MYSQL_QUOTA_FIELD quota
```

Then restart Courier:

```
chkconfig --levels 235 courier-imap on  
/etc/init.d/courier-authlib restart  
/etc/init.d/courier-imap restart
```

When `courier-imap` starts for the first time, the application will automatically create the authentication file `/usr/lib/courier-imap/share/imapd.pem` and `usr / lib / courier-imap / share / pop3d.pem` from the file `/usr / lib / courier-imap / etc / imapd.cnf` and `/usr/lib/courier-imap/etc/pop3d.cnf` . Because the `.cnf` file contains the parameter line `CN = localhost`, but our server is named `server1.example.com`, and the authentication process will have problems when using the TLS connection. To resolve this, delete both files above:

```
cd /usr/lib/courier-imap/share
rm -f imapd.pem
rm -f pop3d.pem
```

and replace the line `CN = localhost` in `/usr/lib/courier-imap/etc/imapd.cnf` and `/usr/lib/courier-imap/etc/pop3d.cnf` with `CN = server1.example.com` :

```
vi /usr/lib/courier-imap/etc/imapd.cnf
```

```
[.]
```

```
CN = server1.example.com
```

```
[.]
```

```
en /usr/lib/courier-imap/etc/pop3d.cnf
```

```
[.]
```

```
CN = server1.example.com
```

```
[.]
```

Then re-create both authentication files:

```
./mkimapdcert
```

```
./mkpop3dcert
```

restart restart courier-authlib and courier-imap:

```
/etc/init.d/courier-authlib restart
```

```
/etc/init.d/courier-imap restart
```

Run the command:

```
telnet localhost pop3
```

to check if the POP3 server is working properly. The returned result will look like + OK Hello there (type quit to return to the Linux shell)

```
[root @ server1 share] # telnet localhost pop3
```

```
Trying :: 1 .
```

```
Connected to localhost.
```

```
Escape character is '^'.
```

```
+ OK Hello there.
```

```
thoát
```

```
+ OK Better luck next time.
```

```
Connection closed by foreign host.
```

```
[root @ server1 share] #
```

Edit / etc / aliases

Now, we will open the file / etc / aliases. Make sure the postmaster variable points to the root account like this:

```
vi / etc / aliases
```

```
[.]  
postmaster: root  
root: postmaster@yourdomain.tld  
[.]
```

or like this (if it is an administrator account):

```
[.]  
postmaster: root  
root: administrator  
[.]
```

Whenever editing / etc / aliases, run the following command:

```
newaliases
```

and then restart Postfix:

```
/etc/init.d/postfix restart
```

Install Amavisd-new, SpamAssassin and ClamAV

To install these applications, use the following command:

```
yum install amavisd-new spamassassin clamav clamav-data clamav-server clamav-update unzip bzip2
```

Next, edit the /etc/amavisd/amavisd.conf file:

```
/etc/amavisd/amavisd.conf
```

In this file, we will modify the 5 parameters:

First, change:

```
$ mydomain = 'example.com'; # m?t m?c ??nh m?c ??nh cho các ph?n khác
```

into:

```
$ mydomain = 'localhost';  
# $ mydomain = 'example.com'; # m?t m?c ??nh m?c ??nh cho các ph?n khác
```

Second, change:

```
$ sa_tag_level_deflt = 2.0; # add spam headers if at, or above mà c?p  
$ sa_tag2_level_deflt = 6.2; # add 'spam detected' headers at that level  
$ sa_kill_level_deflt = 6.9; # triggers spam evasive actions (eg kh?i th?)  
$ sa_dsn_cutoff_level = 10; # spam level ? sau m?t s? DSN không ???c g?i
```

into:

```
$ sa_tag_level_deflt = 2.0; # add spam headers if at, or above mà c?p  
$ sa_tag2_level_deflt = 4.0; # add 'spam detected' headers at that level  
$ sa_kill_level_deflt = $ sa_tag2_level_deflt; # triggers spam evasive actions (eg kh?i th?)  
$ sa_dsn_cutoff_level = 10; # spam level ? sau m?t s? DSN không ???c g?i
```

```
# $ sa_tag_level_deflt = 2.0; # add spam headers if at, or above mà c?p  
# $ sa_tag2_level_deflt = 6.2; # add 'spam detected' headers at that level  
# $ sa_kill_level_deflt = 6.9; # triggers spam evasive actions (eg kh?i th?)  
# $ sa_dsn_cutoff_level = 10; # spam level ? sau m?t s? DSN không ???c g?i
```

Note: adjusting the spam score parameter at will

Tuesday, change:

```
# @lookup_sql_dsn =  
# ['DBI: mysql: database = mail; host = 127.0.0.1; port = 3306', 'user1', 'passwd1'],  
# ['DBI: mysql: database = mail; host = host2', 'username2', 'password2'],  
# ["DBI: SQLite: dbname = $ MYHOME / sql / mail_prefs.sqlite", "", "];  
# @storage_sql_dsn = @lookup_sql_dsn; # none, same, or separate database
```

into:

```
# @lookup_sql_dsn =  
# ['DBI: mysql: database = mail; host = 127.0.0.1; port = 3306', 'user1', 'passwd1'],  
# ['DBI: mysql: database = mail; host = host2', 'username2', 'password2'],  
# ["DBI: SQLite: dbname = $ MYHOME / sql / mail_prefs.sqlite", "", "];  
# @storage_sql_dsn = @lookup_sql_dsn; # none, same, or separate database
```

```
@lookup_sql_dsn =  
(['DBI: mysql: database = mail; host = 127.0.0.1; port = 3306', 'mail_admin', 'mail_admin_password']);
```

```
$ sql_select_policy = 'SELECT "Y" as local FROM domains WHERE CONCAT ("@", domain) IN (% k);
```

```
$ sql_select_white_black_list = undef; # white undef disables SQL white / blacklisting
```

```
$ recipient_delimiter = '+'; # (default is '+')  
$ replace_existing_extension = 1; # (default is false)  
$ localpart_is_case_sensitive = 0; # (default is false)
```

Next, change:

```
# $ recipient_delimiter = '+'; # undef disables address extensions altogether  
# khi ???c phép thêm các t?p tin thêm thêm, c?ng thêm Postfix / main.cf: recipient_delimiter = +
```

into:

```
$ recipient_delimiter = undef; # undef disables address extensions altogether  
# $ recipient_delimiter = '+'; # undef disables address extensions altogether  
# khi ???c phép thêm các t?p tin thêm thêm, c?ng thêm Postfix / main.cf: recipient_delimiter = +
```

Finally, change:

```
$ final_virus_destiny = D_DISCARD;  
$ final_banned_destiny = D_BOUNCE;  
$ final_spam_destiny = D_DISCARD;  
$ final_bad_header_destiny = D_BOUNCE;
```

into:

```
$ final_virus_destiny = D_REJECT;  
$ final_banned_destiny = D_REJECT;  
$ final_spam_destiny = D_PASS;  
$ final_bad_header_destiny = D_PASS;  
  
# $ final_virus_destiny = D_DISCARD;  
# $ final_banned_destiny = D_BOUNCE;  
# $ final_spam_destiny = D_DISCARD;  
# $ final_bad_header_destiny = D_BOUNCE;
```

After applying the change, the /etc/amavisd/amavisd.conf file will look like this:

[**view command**]

In it, amavisd-new is an application that integrates Postfix and SpamAssassin / ClamAV together. When installing ClamAV, a work order was set up to update the identity database for ClamAV every 3 hours. But the feature only works when we activate inside / etc / sysconfig / freshclam and /etc/freshclam.conf:

en / etc / sysconfig / freshclam

and annotate outside the FRESHCLAM_DELAY line at the end as follows:

```
## When changing the periodicity of freshclam runs in crontab,  
## this value must be adapt also. Its value là th?i gian gi?a  
## hai freshclam theo sau trong trong các các các các ch??ng trình này Eg for the default  
##  
## | 0 * / 3 * * * .  
##  
## crontab line, the value is 180 (minutes).  
# FRESHCLAM_MOD =  
  
## A value predefined for delay in seconds. By default, giá tr? là  
## Calcul ???c b?i ch??ng trình 'hostid'. Không tìm th?y giá tr? này  
## times timespans of 3 hours between two subsequent freshclam runs.  
##  
## This option accepts hai các giá tr?:  
## 'disabled-warn' . disables the automatic freshclam update and  
## gives out a warning  
## 'disabled' . disables the automatic freshclam silently  
# FRESHCLAM_DELAY =  
  
### !!!!! REMOVE ME !!!!!  
### REMOVE ME: By default, the freshclam update is disabled to avoid  
### REMOVE ME: network access without prior activation  
# FRESHCLAM_DELAY = disabled-warn # REMOVE ME  
en /etc/freshclam.conf
```

caption outside the Example line:

```
[.]  
# Comment hay g? b? dòng d??i.  
#Example  
[.]
```

Next step, create boot paths for ClamAV and amavisd-new, update ClamAV and launch both services:

```
chkconfig --levels 235 amavisd on  
chkconfig --levels 235 clamd.amavisd on  
/usr/bin/freshclam  
/etc/init.d/amavisd start  
/etc/init.d/clamd.amavisd start
```

Next, configure Postfix to be able to send mail through amavisd-new:

```
postconf -e 'content_filter = amavis: [127.0.0.1]: 10024'  
postconf -e 'receive_override_options = no_address_mappings'
```

Then add the following command line to /etc/postfix/master.cf:

```
en /etc/postfix/master.cf  
  
[.]  
amavis unix - - - - 2 smtp  
-o smtp_data_done_timeout = 1200  
-o smtp_send_xforward_command = yes  
  
127.0.0.1:10025 inet n - - - - smtpd  
-o content_filter =  
-o local_recipient_maps =  
-o relay_recipient_maps =  
-o smtpd_restriction_classes =  
-o smtpd_client_restrictions =  
-o smtpd_helo_restrictions =  
-o smtpd_sender_restrictions =  
-o smtpd_recipient_restrictions = permit_mynetworks, reject  
-o mynetworks = 127.0.0.0 / 8  
-o strict_rfc821_envelopes = yes  
-o receive_override_options = no_unknown_recipient_checks, no_header_body_checks  
-o smtpd_bind_address = 127.0.0.1
```

Restart Postfix to apply the changes:

```
/etc/init.d/postfix restart
```

Install Razor, Pyzor and DCC and configure SpamAssassin

Razor, Pyzor and DCC are email filters with interactive features. To install Razor and Pyzor, type the following command:

```
yum install perl-Razor-Agent pyzor
```

Initialize both services:

```
chmod -R a + rX /usr/share/doc/pyzor-0.5.0 /usr/bin/pyzor /usr/bin/pyzord  
chmod -R a + rX /usr/lib/python2.6/site-packages/pyzor  
su -m amavis -c 'pyzor --homedir /var/spool/amavisd discover'  
su -m amavis -c 'razor-admin -home=/var/spool/amavisd -create'  
su -m amavis -c 'razor-admin -home=/var/spool/amavisd -register'
```

Và cài ??t DCC theo cách sau:

```
cd /tmp
wget http://www.dcc-servers.net/dcc/source/dcc-dccproc.tar.Z
tar xzvf dcc-dccproc.tar.Z
```

You finished reading the article "**Virtualize user and domain accounts**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.