

НАСТРОЙКА ПОЧТОВОЙ СИСТЕМЫ НА БАЗЕ Postfix + Dovecot + PostgreSQL + Amavisd-new + SpamAssassin + ClamAV



В данной статье будет рассмотрена установка почтового сервера на базе AltLinux Master 2.2 с использованием пакетов из репозитория Sisyphus. Наша система обеспечит поддержку виртуальных доменов в связке Postfix + Dovecot + PgSQL с защитой от спама и вирусов с помощью SpamAssassin и ClamAV.

РУСТАМ АТНАГУЛОВ

Итак, посмотрим, для чего предназначен каждый из этих компонентов

- Postfix (<http://postfix.org>) – это популярный MTA (Mail Transfer Agent), позволяющий эффективно передавать письма адресатам.
- Dovecot (<http://dovecot.org>) – Secure IMAP server. IMAP-сервер, рассчитанный на максимальную безопасность и надежность. Преимуществом этой программы является то, что для поиска в больших файлах применяется бинарный древовидный индекс. Как указывает автор этого сервера, он тестировал его с 367 000 почтовыми учетными записями и не обнаружил проблем. Также программа может обслуживать запросы пользователей с помощью протоколов imap, imaps, pop3, pop3s.
- ClamAV (<http://clamav.sourceforge.net>) – opensource-антивирус.
- SpamAssassin (<http://spamassassin.org>) – почтовый фильтр. Предназначен для борьбы со спамом.
- Amavisd-new (<http://www.ijs.si/software/amavisd>) – высокоэффективный и надежный интерфейс между MTA и одним или несколькими фильтрами содержимого: вирусные сканеры, спам-фильтры (в нашем случае SpamAssassin). Amavisd-new написан на Perl, обеспечивающим высокую надежность, мобильность и ремонтпригодность. Он взаимодействует с MTA через (E) SMTP или LMTP или при использовании программ-посредников.

Сначала устанавливаем все необходимые пакеты из репозитория:

```
[root@mosquit /]# apt-get install postfix postfix-pgsql ↵
postgresql postgresql-server dovecot amavisd ↵
clamav spamassassin
```

Далее переходим к конфигурированию нашей почтовой системы. Создаем пользователя, от имени которого будет работать наша почтовая система:

```
[root@mosquit /]# groupadd -g 5000 mailuser
[root@mosquit /]# adduser -d /var/spool/mail -M -g mailuser ↵
-s /sbin/nologin -u 5000 mailuser
```

PostgreSQL

Настройку и конфигурирование PostgreSQL не привожу, т.к. информации об этом достаточно.

Создадим пользователя mailuser в базе данных PostgreSQL, с правами которого Postfix и Dovecot будут подключаться к базе данных и выполнять запросы:

```
[root@mosquit /]# createuser -U postgres
```

```
Enter name of user to add: mailuser
Shall the new user be allowed to create databases? (y/n) n
Shall the new user be allowed to create more new users? (y/n) n
CREATE USER
```

Устанавливаем пароль пользователю mailuser:

```
[root@mosquit /]# psql -U postgres template1
```

```
Welcome to psql 7.3.4, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
template1=# alter user mailuser with password 'topsecret';
ALTER USER
template1=#\q
```

```
/var/lib/postgresql/data/pg_hba.conf
```

#TYPE	DATABASE	USER	IP-ADDRESS	IP-MASK	METHOD
local	all	all			password
host	all	all	127.0.0.1	255.255.255.255	password

Следующий этап – это создание базы данных, в которой будет все храниться.

```
[root@mosquit /]# createdb -U postgres mails
```

```
CREATE DATABASE
```

База данных mails создана. Далее создаем в ней таблицы. Есть 2 способа: писать SQL-команды руками в клиенте postgresql или приготовить файл с SQL-запросом, а потом создать структуру таблиц, выполнив этот самый запрос. Я выбираю последний способ. Создаем файл createtables.sql:

```
CREATE TABLE "public"."users" (
  "userid" VARCHAR(128) NOT NULL,
  "password" VARCHAR(128),
  "realname" VARCHAR(128),
  "uid" INTEGER NOT NULL,
  "gid" INTEGER NOT NULL,
  "home" VARCHAR(128) NOT NULL,
  "mail" VARCHAR(255),
  CONSTRAINT "users_pkey" PRIMARY KEY("userid")
) WITHOUT OIDS;
```

```
grant select on users to mailuser;
```

```
CREATE TABLE "public"."aliases" (
  "alias" VARCHAR(128) NOT NULL,
  "dest" VARCHAR(128) NOT NULL,
  "comment" TEXT DEFAULT ''::text
) WITH OIDS;
```

```
grant select on aliases to mailuser;
```

Структура таблиц создана:

```
[root@mosquit /]# /usr/bin/psql -U postgres mails < createtable.sql
```

Запускаем PostgreSQL:

```
[root@mosquit /]# service postgresql start
```

Postfix

Переходим к правке конфигурационного файла main.cf, который находится в /etc/postfix.main.cf. Он хорошо комментирован, поэтому внимательно вчитываемся в описание и настраиваем согласно своим требованиям. У меня main.cf выглядит так:

```
main.cf:
# Задаем имя нашего почтового узла
myhostname = testdomain.ru
# Имя нашего домена, если не указывать, то по умолчанию
# выставится минус первый компонент имени узла
mydomain = testdomain.ru
myorigin = $myhostname
# RECEIVING MAIL
# Здесь задаем, на каких сетевых интерфейсах будет работать
# Postfix, в моем случае на всех
inet_interfaces = all
# Список доменов, на которые будет осуществляться доставка
# через local transport
mydestination = localhost, $myhostname,
  localhost.$mydomain, $config_directory/mydestination

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
# Здесь мы указываем, что пользователей testdomain.ru нужно
# просматривать в $virtual_mailbox_maps
local_recipient_maps = $virtual_mailbox_maps
unknown_local_recipient_reject_code = 550

# TRUST AND RELAY CONTROL
# здесь указываем, кому разрешен relay, т.е. пользователям
# обслуживаемой сети
mynetworks = 127.0.0.0/8, 10.70.1.0/24
# Задаем директорию, где будет храниться почта
mail_spool_directory = /var/spool/mail
mailbox_command = /usr/bin/procmail -a $DOMAIN -d $LOGNAME
# Наш почтовый сервер требует сначала команду helo
smtpd_helo_required = yes

transport_maps = hash:/etc/postfix/transport.cf
# Задаем директорию, где будет храниться почта
# из виртуальных доменов
virtual_mailbox_base = /var/spool/mail/
# Просмотр таблиц с целью проверки действительности адреса
virtual_mailbox_maps = pgsq:/etc/postfix/mailbox.pgsq
virtual_alias_maps = pgsq:/etc/postfix/aliases.pgsq
# Использую статические uid & gid; 5000 - это mailuser
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
virtual_minimum_uid = 5000

# INSTALL-TIME CONFIGURATION INFORMATION
readme_directory = /etc/postfix/README_FILES
sample_directory = /etc/postfix/samples
sendmail_path = /usr/sbin/sendmail
setgid_group = postdrop
command_directory = /usr/sbin
manpage_directory = /usr/share/man
daemon_directory = /usr/lib/postfix
newaliases_path = /usr/bin/newaliases
mailq_path = /usr/bin/mailq
```

```
queue_directory = /var/spool/postfix
mail_owner = postfix
# Этот параметр относится к Amavisd
# Пока комментируем, чтобы можно было проверить
# работоспособность postfix.
#content_filter=smt-amavis:[127.0.0.1]:10024
```

Остальные параметры по умолчанию, но их можно изменить по собственному усмотрению.

Далее редактируем конфигурационный файл master.cf:

```
master.cf:
#=====
# service type Type Private Unpriv Chroot wakeup maxproc command + args
# (yes) (yes) (yes) (yes) (never) (100)
#=====
smtp inet n - - - - smtpd
pickup fifo n - - 60 1 pickup
cleanup unix n - - - 0 cleanup
qmgr fifo n - - 300 1 qmgr
rewrite unix - - - - - trivial-rewrite
bounce unix - - - - 0 bounce
defer unix - - - - 0 bounce
flush unix n - - 1000 0 flush
proxymap unix - - n - - proxymap
smtp unix - - - - - smtp
relay unix - - - - - smtp
showq unix n - - - - showq
error unix - - - - - error
local unix - n n - - local
virtual unix - n n - - virtual
lmtp unix - - - - - lmtp

# далее относится к Amavisd
smt-amavis unix - - n - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
127.0.0.1:10025 inet n - n - - smtpd
-o content_filter=
-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 smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
```

И соответственно создаем файлы aliases.pgsq, mailbox.pgsq, transport.cf:

```
aliases.pgsq:
hosts = localhost
user = mailuser
password = topsecret
dbname = mails
table = aliases
select_field = dest
where_field = alias

mailbox.pgsq:
hosts = localhost
user = mailuser
password = topsecret
dbname = mails
table = users
select_field = home
where_field = userid

transport.cf:
testdomain.ru virtual:
another.domain.ru virtual:
```

В transport.cf указываем, каким образом будут обрабатываться домены, наши домены – через virtual, в противном случае они будут обработаны с помощью local_transport.

Запускаем postfix:

```
[root@mosquit /]# service postfix start
```

Проверяем работу нашего сервера:

```
[root@mosquit /]# telnet 127.0.0.1 25
```

```
Trying 127.0.0.1...
Connected to 10.70.1.254.
Escape character is '^]'.
helo localhost
220 testdomain.ru ESMTP Postfix
250 testdomain.ru
quit
```

Работает, иначе смотрим лог-файлы в /var/log/maillog.

Создание учетных записей пользователей

Для этого я написал пару простейших скриптов на bash:

```
add mailuser.bash:
#!/bin/bash
basedir=/var/spool/mail

echo "Введите e-mail в виде name@domain"
read email
pos=`expr match "$email" '[a-z1-9A-Z.]*@'`
if [ $pos = "0" ]
then echo "Неправильное имя пользователя"
exit

fi
domain=${email:$pos}
name=${email:0:($pos-1)}
cd $basedir/$domain/$name &> /dev/null
if [ $? = "0" ];
then
echo 'Такой пользователь существует !! '
exit

fi
cd $basedir/$domain &> /dev/null
if [ $? = "1" ];
then
echo 'Такого домена не существует'
exit

fi
echo "Введите полное имя "
read fullname
echo "Введите пароль "
read pass
echo "пароль $pass"
echo -E "INSERT INTO public.users (userid, password, \
realname, uid, gid, home, mail) VALUES ('$email', '$pass', \
'$fullname', '42', '42', '$domain/$name/inbox', NULL);" \
| /usr/bin/psql -U postgres mails
mkdir $basedir/$domain/$name
touch $basedir/$domain/$name/inbox
chmod 600 $basedir/$domain/$name/inbox
chmod 700 $basedir/$domain/$name
chown postfix:postfix $basedir/$domain/$name -R
ls $basedir/$domain/$name/inbox &> /dev/null
if [ $? = "1" ];
then
echo 'Ошибка вышла! '
exit

fi
echo "Пользователь создан успешно"

del mailuser.bash:
#!/bin/bash
basedir=/var/spool/mail

echo "Введите e-mail, который хотите удалить в виде name@domain"
```

```
read email
pos=`expr match "$email" '[a-z1-9A-Z.]*@'`
if [ $pos = "0" ]
then echo "Неправильное имя пользователя"
exit

fi
domain=${email:$pos}
name=${email:0:($pos-1)}
cd $basedir/$domain/$name &> /dev/null
if [ $? = "1" ];
then
echo 'Такой пользователь не существует !! '
exit

fi
echo "DELETE from public.users where userid='$email';" | \
usr/bin/psql -U postgres mails
rm -rf $basedir/$domain/$name &> /dev/null
cd $basedir/$domain/$name &> /dev/null
echo $?
if [ $? = "0" ];
then
echo 'Ошибка вышла! '
exit

fi
echo "Пользователь удален успешно"

add alias.bash:
#!/bin/bash

echo "Введите aliasname в виде name@domain"
read aliasname
pos=`expr match "$aliasname" '[a-z1-9A-Z.]*@'`
if [ $pos = "0" ]
then echo "Неправильный aliasname"
exit

fi

echo "Введите destination в виде name@domain"
read dest
pos=`expr match "$dest" '[a-z1-9A-Z.]*@'`
if [ $pos = "0" ]
then echo "Неправильный destination"
exit

fi

echo "Введите комментарий"
read comment
echo -E "INSERT INTO public.aliases (email, alias, comment) \
VALUES ('$aliasname', '$dest', '$comment');" \
| /usr/bin/psql -U postgres mails
echo "Alias создан успешно"
```

Скрипты созданы на скорую руку и поэтому не совершенны, но всегда существует возможность модифицировать их по своему усмотрению.

Создаем почтового пользователя test@testdomain.ru:

```
[root@mosquit /]# add_mailuser.bash
```

Выполняем инструкции, которые выводятся на экран. Далее проверяем, как Postfix доставляет почту:

```
[root@mosquit /]# service postfix restart
[root@mosquit /]# telnet 127.0.0.1 25
```

```
Trying 127.0.0.1...
Connected to 10.70.1.254.
Escape character is '^]'.
helo localhost
220 testdomain.ru ESMTP Postfix
250 testdomain.ru
mail from: vasia@gde-to.tam
250 Ok
rcpt to: test@testdomain.ru
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
test message
.
```

Обращаю внимание на точку с новой строки после test

message, это означает, что передача данных завершена.

Далее идем в каталог /var/spool/mail/testdomain/test и смотрим содержимое файла inbox, в нем и должно находиться наше сообщение. Если файл пуст, то опять же смотрим лог-файл (maillog).

ClamAV

Идем в /etc/clamav.conf и находим строчку ScanMail и раскомментируем ее, для того чтобы антивирус проверял файлы формата mailbox. Запускаем ClamAV:

```
[root@mosquit ~]# service clamd start
```

Amavisd-new

Далее настраиваем /etc/amavis/amavisd.conf по своему усмотрению, конечно же, перед этим прочитав документацию. Лично я ничего не изменял, но скорее всего со временем придется внести коррективы.

Запускаем Amavisd-new:

```
[root@mosquit ~]# service amavisd start
```

Проверяем его работу:

```
[root@mosquit ~]# telnet 127.0.0.1 10024
```

```
220 [127.0.0.1] ESMTP amavisd-new service ready
quit
```

Раскомментируем строчки: filter=smtp-amavis:[127.0.0.1]:10024 в файле main.cf. Перезапускаем Postfix, запускаем ClamAV, Amavisd:

```
[root@mosquit ~]# service postfix restart
```

Как и ранее, проверяем с помощью telnet работоспособность postfix. Если все нормально, то переходим к настройке Dovecot.

Dovecot

Далее редактируем файл настроек dovecot/etc/dovecot.conf. Так же внимательно вчитываемся в комментарии к параметрам.

```
dovecot.conf:
# Протоколы, которые мы используем
protocols = imap imaps pop3 pop3s
## IMAP login process
login = imap

## POP3 login process
login = pop3

## Mail processes

first_valid_uid = 5000
# Путь к директории, где хранится почта, где %d -
# имя домена, %n - имя пользователя
default_mail_env = mbox:/var/spool/mail/%d/%n/: J
INDEX=/var/spool/mail/%d/%n

mbox_locks = fcntl flock
## Authentication processes
auth = default
auth_mechanisms = plain
auth_userdb = pgsqldb /etc/dovecot/pgsqldb.conf
auth_passdb = pgsqldb /etc/dovecot/pgsqldb.conf
# root т.к. использовать порты до 1024 порта может только root
```

```
auth_user = root
```

После этого нам требуется сгенерировать SSL-сертификат, т.к. у нас заявлены протоколы imap и pop3s (s – secure).

Для этого правим файл /etc/dovecot/dovecot-openssl.cnf:

```
[ req ]
default_bits = 1024
encrypt_key = yes
distinguished_name = req_dn
x509_extensions = cert_type
prompt = no

[ req_dn ]
# country (2 letter code)
C=RU

# State or Province Name (full name)
ST=Bashkortostan

# Locality Name (eg. city)
L=Ufa

# Organization (eg. company)
O=Home Organizations.

# Organizational Unit Name (eg. section)
OU=POP3 IMAP server

# Common Name (*.example.com is also possible)
CN=testdomain.ru

# E-mail contact
emailAddress=master@testdomain.ru

[ cert_type ]
nsCertType = server
```

Запускаем скрипт для генерации сертификата.

```
[root@mosquit ~]# /usr/share/dovecot/mkcert.sh
```

Правим файл /etc/dovecot/pgsqldb.conf, в котором указываем, как Dovecot будет подключаться и забирать данные.

```
connect = host = localhost dbname = mails user = mailuser J
password = secret
default_pass_scheme = PLAIN
password_query = SELECT password FROM users WHERE userid = '%u'
user_query = SELECT '/var/spool/mail/' || home, uid, gid J
FROM users WHERE userid = '%u'
```

Все настройки произведены, можно испытать работоспособность нашего pop3-сервера:

```
[root@mosquit ~]# service dovecot start
```

Настраиваем наш почтовый клиент и проверяем, доставляется почта или нет. Имя пользователя указывается полное, т.е. test@testdomain.ru, а не test.

Если же не получается, то заглядываем в /var/log/messages и устраняем причину (довольно часто причиной бывает банальная опечатка при редактировании конфигурационных файлов).

Используемая литература:

1. Postfix Documentation: <http://postfix.org/documentation.html>
2. Documentation: <http://dovecot.org/documentation.html>
3. DovecotPostgresql: <http://wiki.dovecot.org/moint.cgi/DovecotPostgresql>