Расширенное администрирование Linux

Блок 4

v 1.02

Оглавление

Система электронной почты

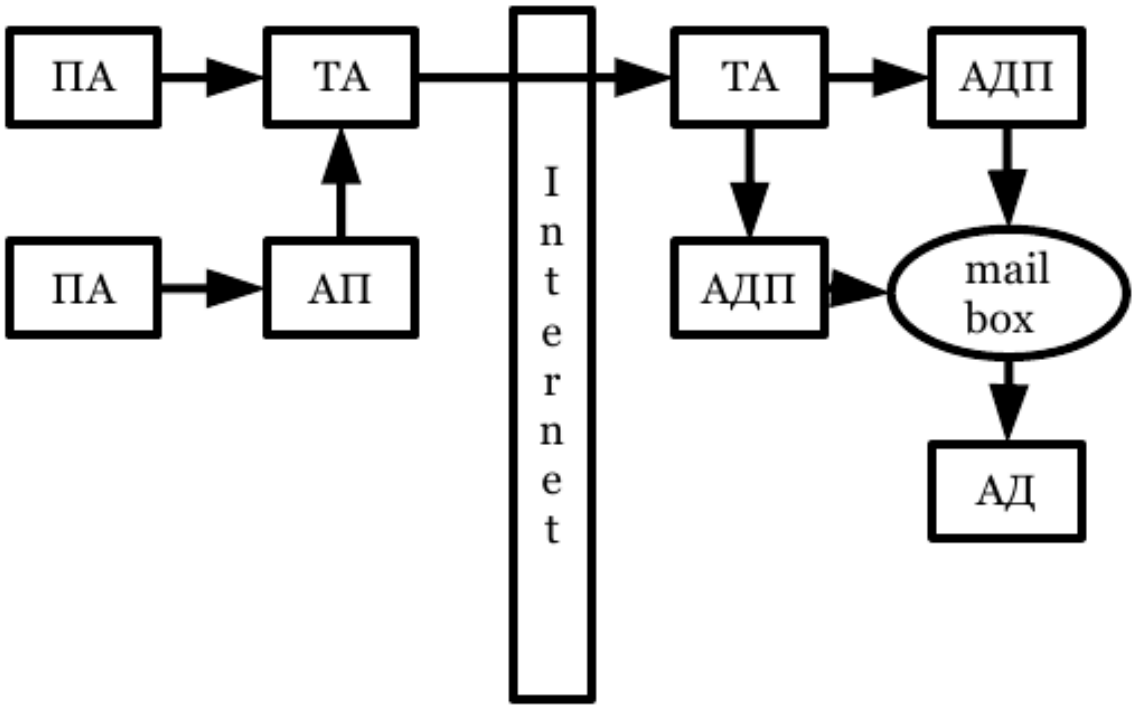
Появление электронной почты можно отнести к 1965 году, когда сотрудники Массачусетского технологического института (MIT) Ноэль Моррис и Том Ван Влек написали программу MAIL для операционной системы CTSS (Compatible Time-Sharing System), установленную на компьютере IBM 7090/7094.

Общее развитие электронной почты шло через развитие локального взаимодействия пользователей на многопользовательских системах. Пользователи могли, используя программу mail (или её эквивалент), пересылать друг другу сообщения в пределах одного мейнфрейма (компьютера). Следующий шаг был в возможности переслать сообщение пользователю на другой машине — для этого использовалось указание имени машины и имени пользователя на машине. Адрес мог записываться в виде foo!joe (пользователь joe на компьютере foo). Третий шаг для становления электронной почты произошёл в момент появления передачи писем через третий компьютер. В случае использования UUCP адрес пользователя включал в себя маршрут до пользователя через несколько промежуточных машин (например, gate1!gate2!foo!joe — письмо для joe через машину gate1, gate2 на машину foo). Недостатком такой адресации было то, что отправителю (или администратору машины, на которой работал отправитель) необходимо было знать точный путь до машины адресата.

После появления распределённой глобальной системы имён DNS, для указания адреса стали использоваться доменные имена — user@example.com — пользователь user на машине example.com. Одновременно с этим происходило переосмысление понятия «на машине»: для почты стали использоваться выделенные сервера, на которые не имели доступ обычные пользователи (только администраторы), а пользователи работали на своих машинах, при этом почта приходила не на рабочие машины пользователей, а на почтовый сервер, откуда пользователи забирали свою почту по различным сетевым протоколам (среди распространённых на настоящий момент — POP3, IMAP, MAPI, веб-интерфейсы). Одновременно с появлением DNS была продумана система резервирования маршрутов доставки почты, а доменное имя в почтовом адресе перестало быть именем конкретного компьютера и стало просто почтовым доменом, за обслуживание которого могли отвечать многие сервера (возможно, физически размещённые на разных континентах и в разных организациях).

Кроме того, существовали (и существуют по настоящий момент) и другие системы электронной почты (некоторые из них существуют и сейчас), как-то Netmail в сети FidoNET, X.400 в сетях X.25. Доступ к ним из интернет и обратно осуществляется через почтовый шлюз. Для маршрутизации почты в сетях X.25 в DNS предусмотрена специальная ресурсная запись с соответствующим названием X25 (код 19).

Компоненты электронной почты



Система электронной почты состоит из нескольких компонентов. Они могут быть выполнены в виде одной или нескольких программ.

Можно выделить следующие компоненты:

- Пользовательский агент (ПА)
- Транспортный агент (ТА)
- Агент подачи почты (АП)
- Агент доставки почты (АДП)
- Агент доступа (АД)

Пользовательский агент

При помощи пользовательских агентов пользователи составляют и отправляют письма. Агент должен сформировать тело письма согласно стандарта и передать его на отправку транспортному агенту или агенту подачи.
В роли пользовательских агентов выступают такие программы как: The Bat, Outlook и Outlook Express. Если говорить про Linux: Evolution, KMail, pine и др.
При передачи письма транспортному агенту и агенту подачи используется протокол SMTP.

Транспортный агент

Транспортный агент выполняет две основные задачи:

- прием почты от пользовательского агента и пересылка ее на другой транспортный агент
- прием почты от других транспортных агентов

При приеме почты от пользователя он должен проверить правильность адреса назначения, возможность доставки почты и доставить почту по назначению.

На другой стороне транспортный агент проверяет: предназначено ли это письмо для данной машины (домена), есть ли почтовый ящик пользователя на машине. После проверок он принимает письмо и передает его Агенту доставки почты.
В мире UNIX существует большое количество программ, реализующих функции транспортного агента. Среди наиболее популярных бесплатных реализаций транспортных агентов можно выделить sendmail, postfix, exim и qmail.

Агент подачи почты

Агенты подачи почты — это одна из разновидностей режима работы транспортного агента. Агенты подачи применяются на почтовых узлах с напряженным трафиком. Его задача облегчить работу основного транспортного агента.
Агент подачи:

- Проверяет, являются ли имена узлов полностью определенными.
- Модифицирует заголовки сообщений, полученных от неправильно работающих пользовательских агентов.
- Проверяет все ошибки перед передачей письма транспортному агенту.

Агент подачи слушает запросы на 587 порту, поэтому все пользовательские агенты необходимо сконфигурировать таким образом, чтобы они отправляли почту на этот порт. Все особенности работы агента подачи описаны в RFC 2476.

Агент доставки почты

Транспортный агент после получения почты сам не доставляет ее в почтовый ящик пользователя. Он передает ее агенту доставки почты, задача которого доставить почту в почтовый ящик пользователя.
В качестве агента доставки может выступать простейшая программа, которая просто складывает почту. Существуют и более сложные программы, которые при доставке почты могут осуществлять ее фильтрацию, например, procmail. Для чего предназначены программы доставки почты описано в RFC2476.

Агенты доступа

Агенты доступа позволяют пользователю получить доступ к своему почтовому ящику. Они выполнены в виде программ, организующих доступ к почтовому ящику по протоколу pop или imap.
Если пользователь работает локально на машине, на которой хранятся его почтовые ящики, он может получить доступ без агента доступа, обращаясь к ним на прямую. По такому принципу работают mail и pine.

Протокол SMTP

В качестве основного протокола взаимодействия в системе электронной почты используются протоколы SMTP (Simple Mail Transfer Protocol) и ESMTP (Extended SMTP). Они описаны в RFC2821, 1869,1870,1891 и 1985.
Протокол SMTP задумывался как простой протокол взаимодействия, при помощи которого пользователь мог напрямую общаться с почтовым транспортным агентом. Конечно же, сейчас пользователи сами не работают с транспортными агентами. Для облечения работы используются пользовательские агенты. Но они (пользовательские агенты) для взаимодействия с транспортными агентами используют протокол SMTP или ESMTP.
В качестве примера можно показать, как пользователь при помощи программы telnet может подключиться к транспортному агенту и отправить письмо.
Пользователь, при помощи транспортного агента может только отправлять письма. Прием почты происходит при помощи агентов доступа.
Для подключения к почтовому транспортному агенту использовалась программа telnet, с явным указанием прота 25.

```
$ telnet localhost 25
Trying 127.0.0.1...
Connected to ubuntu-server.
Escape character is '^]'.
220 localhost ESMTP Postfix
```

При отправке почты обязательно следует указать от кого эта почта отправляется. Для этого используется команда mail from про- токола SMTP. Транспортный агент по умолчанию требует в адресе отправителя обязательное указание домена в e-mail отправителя.

```
mail from: sergio
250 2.1.0 Ok
```

Если с почтовым сообщением отправителя все в порядке, транспортный агент выдает подтверждение. Теперь при помощи команды rcpt to указывается адрес получателя.

```
rcpt to: root
250 2.1.5 root... Recipient ok
```

При помощи команды data начинается ввод текста письма. Последняя строка должна содержать единственный символ «.», который обозначает конец письма.

```
data
354 End data with <CR><LF>.<CR><LF>
Hello world!
Test mail
.
```

После ввода символа «.» транспортный агент принимает письмо на доставку, о чем выдается соответствующее сообщение.

```
250 2.0.0 Ok: queued as  75B5726911
```

Для завершения сеанса связи используется команда quit.

```
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Как видно из приведенного примера, транспортному агенту для доставки почты необходимо указать два параметра: адрес отправителя и адрес получателя, а также тело

письма.

Заголовки, которые Вы видите в теле письма, например, From:, To:, X-Mailer: и т.д., транспортным агентом рассматриваются как тело письма и напрямую не обрабатываются. Хотя sendmail можно настроить таким образом, чтобы он проверял или изменял такие заголовки.
При доставке письма транспортный агент добавляет в его заголовок строки, свидетельствующие о прохождении письма через транспортный агент.
Каждый транспортный агент, через который проходит письмо, добавляет похожую строку. По этим строкам можно определить маршрут прохождения письма.
Выше был показан достаточно простой пример взаимодействия сервера и клиента. Не учитывающий шифрование канала и авторизацию пользователя.

Протокол POP3

POP3 (англ. Post Office Protocol Version 3 — протокол почтового отделения, версия 3) используется почтовым клиентом для получения сообщений электронной почты с сервера. Обычно используется в паре с протоколом SMTP. Предыдущие версии протокола (POP, POP2) устарели.
Стандарт протокола POP3 определён в RFC 1939. Расширения и методы авторизации определены в RFC 2195, RFC 2449, RFC 1734, RFC 2222, RFC 3206, RFC 2595. Существуют реализации POP3-серверов, поддерживающие TLS и SSL.
Альтернативным протоколом для сбора сообщений с почтового сервера является IMAP.

Состояния сеанса

В протоколе POP3 предусмотрено 3 состояния сеанса:

Авторизация
: Клиент проходит процедуру Аутентификации

Транзакция
: Клиент получает информацию о состоянии почтового ящика, принимает и удаляет почту

Обновление
: Сервер удаляет выбранные письма и закрывает соединение

Команды протокола

APOP [имя] [digest]

Команда служит для передачи серверу имени пользователя и зашифрованного пароля(digest)

Аргументы
: [имя] - строка, указывающая имя почтового ящика.
[digest]- временная метка, зашифрованная паролем пользователя по алгоритму MD5. В случае поддержки этой команды временная метка получается при соединении с сервером:

+OK POP3 server ready <**1896.698370952@meshach.smallorg.org**>

Ограничения
: Её поддержка не является обязательной

Возможные ответы

- +OK maildrop has n message
- -ERR password suplied for [имя] is incorrect

DELE [сообщение]

Сервер помечает указанное сообщение для удаления. Сообщения, помеченные на удаление, реально удаляются только после закрытия транзакции (закрытие транзакций происходит обычно после посыла команды QUIT, кроме этого, например, на серверах закрытие транзакций может происходить по истечению определенного времени, установленного сервером).

Аргументы
: [сообщение] - номер сообщения.

Ограничения
: Доступна после успешной идентификации

Возможные ответы

- +OK message deleted
- -ERR no such message

LIST [сообщение]

Если был передан аргумент, то сервер выдаёт информацию об указанном сообщении. Если аргумент не был передан, то сервер выдаёт информацию обо всех сообщениях, находящихся в почтовом ящике. Сообщения, помеченные для удаления не перечисляются.

Аргументы
    [сообщение]-номер сообщения (необязательный аргумент)
Ограничения
    Доступна после успешной идентификации
Возможные ответы

- +OK scan listing follows
- -ERR no such message

NOOP

Сервер ничего не делает, всегда отвечает положительно

Аргументы
    Нет.
Ограничения
    Нет.
Возможные ответы

- +OK

PASS [пароль]

Передаёт серверу пароль почтового ящика

Аргументы
    [пароль] - пароль для почтового ящика.
Ограничения
    Работает после успешной передачи имени почтового ящика.
Возможные ответы

- +OK maildrop locked and ready
- -ERR invalid password
- -ERR unable to lock maildrop

RETR [сообщение]

Сервер передаёт сообщение с указанным номером

Аргументы
    [сообщение] - номер сообщения
Ограничения
    Доступна после успешной идентификации
Возможные ответы

- +OK message follows
- -ERR no such message

RSET

Этой командой производится откат транзакций внутри сессии. Например, если пользователь случайно пометил на удаление какие-либо сообщения, он может убрать эти пометки, отправив эту команду

Аргументы
    Нет.
Ограничения
    Доступна после и до успешной идентификации
Возможные ответы

- +OK

STAT

Сервер возвращает количество сообщений в почтовом ящике плюс размер, занимаемыми этими сообщениями на почтовом ящике

Аргументы
    Нет
Ограничения
    Доступна после успешной идентификации
Возможные ответы

- +OK a b

TOP [сообщение] [количество строк]

Сервер возвращает заголовки указанного сообщения, пустую строку и указанное количество первых строк тела сообщения.

Аргументы
    [сообщение] - номер сообщения
    [количество строк] - сколько строк нужно вывести
Ограничения
    Доступна после успешной идентификации
Возможные ответы

- +OK n octets
- -ERR no such message

USER [имя]

Передаёт серверу имя пользователя

Аргументы
    [имя] - строка, указывающая имя почтового ящика.
Ограничения
    Нет.
Возможные ответы

- +OK name is a valid mailbox
- -ERR never heard of mailbox name

QUIT

Аргументы
    Нет.
Ограничения
    Нет.
Возможные ответы

- +OK

Протокол IMAP

IMAP (англ. Internet Message Access Protocol — "Протокол доступа к электронной почте Интернета") — протокол прикладного уровня для доступа к электронной почте. Аналогичен POP3, т.е. служит для работы со входящими письмами, однако обеспечивает дополнительные функции, в частности, возможность провести поиск по ключевому слову, не сохраняя почту в локальной памяти.

IMAP предоставляет пользователю богатые возможности для работы с почтовыми ящиками, находящимися на центральном сервере. Почтовая программа, использующая этот протокол, получает доступ к хранилищу корреспонденции на сервере так, как будто эта корреспонденция расположена на компьютере получателя. Электронными письмами можно манипулировать с компьютера пользователя (клиента) без необходимости постоянной пересылки с сервера и обратно файлов с полным содержанием писем.

Преимущества по сравнению с POP3

IMAP был разработан для замены более простого протокола POP3 и имеет следующие преимущества по сравнению с последним:

- Письма хранятся на сервере, а не на клиенте. Возможен доступ к одному и тому же почтовому ящику с разных клиентов. Поддерживается также *одновременный* доступ нескольких клиентов. В протоколе есть механизмы, с помощью которых клиент может быть проинформирован об изменениях, сделанных другими клиентами.
- Поддержка нескольких почтовых ящиков (или папок). Клиент может создавать, удалять и переименовывать почтовые ящики на сервере, а также перемещать письма из одного почтового ящика в другой.
- Возможно создание общих папок, к которым могут иметь доступ несколько пользователей.
- Информация о состоянии писем хранится на сервере и доступна всем клиентам. Письма могут быть помечены как прочитанные, важные и т.п.
- Поддержка поиска на сервере. Нет необходимости скачивать с сервера множество сообщений для того, чтобы найти одно нужное.
- Поддержка онлайн-работы. Клиент может поддерживать с сервером постоянное соединение, при этом сервер в реальном времени информирует клиента об изменениях в почтовых ящиках, в том числе о новых письмах.
- Предусмотрен механизм расширения возможностей протокола.

Текущая версия протокола имеет обозначение IMAP4rev1 (IMAP, версия 4, ревизия 1). Протокол поддерживает передачу пароля пользователя в зашифрованном виде. Кроме того, IMAP-трафик можно зашифровать с помощью SSL.

Настройка Postfix

Установим поддержку в Postfix Maildir Mailboxes (формат хранения почтовых сообщений, который не требует блокировки файла для поддержания целостности почтового сообщения, потому что сообщение хранится в раздельных файлах с уникальными именами.

Собственно Maildir это директория с тремя поддиректориями с именами tmp new и cur.

Отредактируем для этого main.cf и добавим в него строки:

```
home_mailbox = Maildir/

mailbox_command = procmail -a «$EXTENSION»
```

Перезагрузите Postfix.

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

Добавим поддержку виртуальных почтовых ящиков

Смысл состоит в том, чтобы Postfix использовал учетные записи не системные, а виртуальные, известные ему одному. Это более оправданно из соображений безопасности. Предполагается, что у вас почта будет располагаться как:

```
/home/vmail/unix.specialist.ru/info/
/home/vmail/unix.specialist.ru/sales/
```

Далее идут каталоги, оговариваемые выше, т.е. new, cur, tmp.

Продолжаем править файл main.cf

```
myhostname = localhost
```

Все виртуальные почтовые ящики должны иметь владельца vmail, входящего в группу vmail с uid 5000

```
groupadd -g 5000 vmail
useradd -m -u 5000 -g vmail -s /bin/bash vmail
```

Добавим в main.cf поддержку virtual_mail_box:

```
virtual_mailbox_domains = /etc/postfix/vhosts
virtual_mailbox_base = /home/vmail
virtual_mailbox_maps = hash:/etc/postfix/vmaps
virtual_minimum_uid = 1000
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
```

Создадим файл и добавим в него следующее:

```
nano /etc/postfix/vhosts
```

добавить:

```
unix.specialist.ru
```

Это и есть наши создаваемые почтовые домены.

Создадим файл /etc/postfix/vmaps в него добавим в двух столбцах виртуальные почтовые адреса и место расположения (mailbox) этих адресов в системе:

```
nano /etc/postfix/vmaps

info@unix.specialist.ru unix.specialist.ru/info/
sales@unix.specialist.ru unix.specialist.ru/sales/
```

Конвертируем vmaps в хэш файл

```
postmap /etc/postfix/vmaps
```

Перезагрузим Postfix

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

Пример реально работающего конфига

```
smtpd_banner = $myhostname ESMTP $mail_name
biff = no
append_dot_mydomain = no
myhostname = localhost
home_mailbox = Maildir/
virtual_mailbox_domains = /etc/postfix/vhosts
virtual_mailbox_base = /home/vmail
virtual_mailbox_maps = hash:/etc/postfix/vmaps
virtual_minimum_uid = 1000
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $myhostname
mynetworks = 127.0.0.0/8 192.168.2.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

Проверка виртуальных mailbox

Пошлите письмо info@unix.specialist.ru из консоли:

```
mail info@unix.specialist.ru
```

Проверьте этот же почтовый ящик:

```
cd /home/vmail/unix.specialist.ru/info/new
ls
```

Там должно находиться тестовое письмо

Лабораторная работа

Цель работы

Научиться осуществлять настройку связку postfix+dovecot.

Задачи	*Описание*
1. Базовая настройка	1. Убедитесь, что в вашем DNS-домене присутствует MX-запись **dig mx unix.specialist.ru** 2. Включите поддержку в Postfix Maildir Mailboxes в /etc/postfix/main.cf **home_mailbox = Maildir/** **mailbox_command = procmail -a «$EXTENSION»** 3. Перезагрузите Postfix **/etc/init.d/postfix restart**
2. Настройка виртуальных почтовых ящиков	1. Создайте необходимые директории **/home/vmail/unix.specialist.ru/info/** **/home/vmail/unix.specialist.ru/sales/** 2. Добавьте строку в /etc/postfix/main.cf **myhostname = localhost** 3. Добавьте пользователя и группу vmail **groupadd -g 5000 vmail** **useradd -m -u 5000 -g 5000 -s /bin/bash vmail** 4. Добавим в /etc/postfix/main.cf поддержку virtual_mail_box: **virtual_mailbox_domains = /etc/postfix/vhosts** **virtual_mailbox_base = /home/vmail** **virtual_mailbox_maps = hash:/etc/postfix/vmaps** **virtual_minimum_uid = 1000** **virtual_uid_maps = static:5000** **virtual_gid_maps = static:5000** 5. Создайте файл /etc/postfix/vhosts и добавьте в него строку описывающую ваш домен. **unix.specialist.ru** 6. Создадим /etc/dovecot/dovecot.conf следующего содержания: **#base_dir = /var/run/dovecot** **protocols = imap pop3** **disable_plaintext_auth = no** **shutdown_clients = yes** **log_path = /var/log/dovecot** **info_log_path = /var/log/dovecot.info** **log_timestamp = "%Y-%m-%d %H:%M:%S "** **ssl_disable = yes** **login_dir = /var/run/dovecot/login** **login_chroot = yes** **login_user = dovecot** **login_greeting = Dovecot ready.**

	mail_location = maildir:/home/vmail/%d/%n **mmap_disable = no** **valid_chroot_dirs = /var/spool/vmail** **protocol imap {** **login_executable = /usr/lib/dovecot/imap-login** **mail_executable = /usr/lib/dovecot/imap** **}** **protocol pop3 {** **login_executable = /usr/lib/dovecot/pop3-login** **mail_executable = /usr/lib/dovecot/pop3** **pop3_uidl_format = %08Xu%08Xv** **}** **auth_executable = /usr/lib/dovecot/dovecot-auth** **auth_verbose = yes** **auth default {** **mechanisms = plain digest-md5** **passdb passwd-file {** **args = /etc/dovecot/passwd** **}** **userdb passwd-file {** **args = /etc/dovecot/users** **}** **user = root** **}** 7. Создайте скрипт /usr/sbin/adddovecotuser для добавления пользователей в систему: **echo "$1" > /tmp/user** **user=`cat /tmp/user \| cut -f1 -d "@"`** **domain=`cat /tmp/user \| cut -f2 -d "@"`** **echo -n "$user@$domain" >> /etc/dovecot/users** **echo -n "::5000:5000::" >> /etc/dovecot/users** **echo -n "/home/vmail/" >> /etc/dovecot/users** **echo "$domain/:/bin/false" >> /etc/dovecot/users** **/usr/bin/maildirmake.dovecot /home/vmail/ \ $domain/$user 5000:5000** **echo $1 $domain/$user/ >> /etc/postfix/vmaps** **postmap /etc/postfix/vmaps** **postfix reload** 8. Создайте скрипт /usr/sbin/mkdovecotpasswd для добавления пользователей в систему: **mkpasswd --hash=md5 $2 > /tmp/hash** **echo "$1:`cat /tmp/hash`" >> /etc/dovecot/passwd** 9. Создайте учетные записи: **adddovecotuser info@unix.specialist.ru** **mkdovecotpasswd info@unix.specialist.ru pass1** **adddovecotuser sales@unix.specialist.ru** **mkdovecotpasswd sales@unix.specialist.ru pass2** 10. Защитите файл пароля: **chmod 640 /etc/dovecot/passwd**
3. Запуск сервера	1. Перезапустите postfix-сервер

	/etc/init.d/postfix restart 2. Перезапустите dovecot-сервер **/etc/init.d/dovecot restart**
4. Проверка работы сервера	1. Тестируем postfix **$ telnet localhost 25** **Trying 127.0.0.1...** **Connected to ubuntu-server.** **Escape character is '^]'.** **220 localhost ESMTP Postfix** **mail from: root** **250 2.1.0 Ok** **rcpt to: info@unix.specialist.ru** **250 2.1.5 info... Recipient ok** **data** **354 End data with <CR><LF>.<CR><LF>** **Hello world!** **Test mail** **.** **250 2.0.0 Ok: queued as 75B5726911** **quit** **221 2.0.0 Bye** **Connection closed by foreign host.** 2. Тестируем dovecot ***$ telnet ubuntu-server.unix.specialist.ru 110*** **Trying 192.168.10.5...** **Connected to *ubuntu-server.unix.specialist.ru*.** **Escape character is '^]'.** **+OK dovecot ready.** **User info@unix.specialist.ru** **+OK** **pass пароль** **+OK Logged in.** **quit** **+OK Logging out.**

Защита от спама и проверка на вирусы

Самый большой поток спама распространяется через электронную почту (e-mail). В настоящее время доля вирусов и спама в общем трафике электронной почты составляет по разным оценкам от 70 до 95 процентов[5]. В спаме наиболее часто встречается реклама непопулярных товаров: rolex, viagra и т.п..

Спамеры собирают e-mail адреса с помощью специального робота или вручную (редко), используя веб-страницы, конференции Usenet, списки рассылки, электронные доски объявлений, гостевые книги, чаты… Такая программа-робот способна собрать за час тысячи адресов и создать из них базу данных для дальнейшей рассылки по ним спама. Некоторые компании занимаются только сбором адресов, а базы потом продают. Некоторые компании продают спамерам e-mail адреса своих клиентов, заказавших у них товары или услуги по электронной почте. Есть ещё один способ получить большой список работающих e-mail адресов: адреса сначала генерируются случайным образом по заданным шаблонам (от тысячи до миллиона), а потом просто проверяются специальной программой-валидатором на их валидность (существование).

Сейчас один из основных каналов распространения вирусов — электронная почта. Обычно вирусы в письмах электронной почты маскируются под безобидные вложения: картинки, документы, музыку, ссылки на сайты. В некоторых письмах могут содержаться действительно только ссылки, т.е. в самих письмах может и не быть вредоносного кода, но если открыть такую ссылку то можно попасть на специально созданную страницу, содержащую вирусный код. Некоторые вирусы, попав на компьютер, могут использовать адресную книгу пользователя, для рассылки самого себя.

Spamassassin

SpamAssassin — эффективное средство для фильтрации спама, основанное на взаимодействии ключевых компонентов — оценочного демона, транспортного агента и базы шаблонов писем. SpamAssassin использует Байесовскую фильтрацию, обработку DNSBL, Sender Policy Framework, DomainKeys, DKIM, Razor и другие методы распознавания спама. Является проектом верхнего уровня в Apache Software Foundation.

Установка SpamAssasin:

```
aptitude install spamassassin spamc
```

По умолчанию spamassassin выключен, поэтому следует его включить, для чего надо прописать параметр ENABLED=1 в файле **/etc/default/spamassassin** и перезапустить демон:

```
/etc/init.d/spamassassin restart
```

Clamav

Clam AntiVirus — пакет антивирусного ПО, работающий во многих операционных системах, включая Unix-подобные ОС, OpenVMS, Microsoft Windows и Apple Mac OS X.

Выпускается под GNU General Public License и является свободным программным обеспечением.

Главная цель Clam AntiVirus — интеграция с серверами электронной почты для проверки файлов, прикреплённых к сообщениям. В пакет входит масштабируемый многопоточный демон clamd, управляемый из командной строки сканер clamscan, а также модуль обновления сигнатур по Интернету freshclam.

Возможности Clam AntiVirus:

- управление из командной строки;
- возможность использования с большинством почтовых серверов, включая реализацию milter-интерфейса для почтовой системы;
- сканер в виде библиотеки Си;
- сканирование файлов и почты «на лету»;
- определение свыше 360 000 вирусов, червей, троянов, сообщений фишинга;
- анализ сжатых файлов RAR (2.0, 3.0), Zip, Gzip, Bzip2, MS OLE2, MS Cabinet, MS CHM (сжатый HTML) и MS SZDD;
- поддержка сканирования mbox, Maildir и «сырых» почтовых файлов;
- анализ файлов формата Portable Executable, упакованных UPX, FSG или Petite.

Установка Clamav:

```
aptitude install clamav-daemon clamav clamsmtp
```

Теперь добавим служебных пользователей:

```
adduser clamsmtp clamav
```

Перестартуем наши почтовые сервисы:

```
/etc/init.d/postfix restart
/etc/init.d/clamsmtp restart
/etc/init.d/clamav-daemon restart
```

Связывание Postfix+Spamassassin+Clamav

Наиболее простым способом проверки почты является использование скрипта для ее фильтрации. В первую очередь, надо прописать фильтр в файле /etc/postfix/master.cf

В файле master.cf исправляем строку:

```
smtp     inet  n       -       n       -       -       smtpd
```

на

```
smtp      inet  n       -       n       -       -       smtpd -o
content_filter=myfilter:dummy
```

и добавляем

```
    myfilter       unix    -       n       n       -       -       pipe
        flags=R user=clamav argv=/usr/local/bin/myfilter.sh -f ${sender} -- ${recipient}
```

и собственно пишем скрипт /usr/local/bin/myfilter.sh

```
#!/bin/sh
INSPECT_DIR=/tmp #Каталог куда будут сохраняться письма для сканирования
SENDMAIL="/usr/sbin/sendmail -i"
VIRUSADMIN="info@unix.specialist.ru" # адрес для уведомлениий

EX_TEMPFAIL=75
EX_UNAVAILABLE=69

# строка для запуска spamassassin
FILTER_SPAMC="/usr/bin/spamc -u spamfilter -U /var/run/spamd.sock"

trap "rm -f $INSPECT_DIR/in.$$ $INSPECT_DIR/vr.$$ $INSPECT_DIR/vr1.$$" 0 1 2 3 15

# Проверка на спам
cat | $FILTER_SPAMC > $INSPECT_DIR/in.$$ || { echo Cannot save mail to file; exit
$EX_TEMPFAIL; }

# Проверка на вирусы
/usr/bin/clamscan -v -r --no-summary --stdout ${INSPECT_DIR}/in.$$>$INSPECT_DIR/vr.$$

# Результат проверки
AV_RESULT=$?

case "$AV_RESULT" in
0)
  # Проверено. Мин нет :)
  $SENDMAIL "$@" <${INSPECT_DIR}/in.$$
  exit 0
  ;;
1)
  # Обнаружен вирус. Посылаем уведомление админу
  echo "Subject: VIRUS FOUND" >> $INSPECT_DIR/vr1.$$
  echo >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  echo "* MAIL                                      *" >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  echo >> $INSPECT_DIR/vr1.$$
  # Включаем в отчет реальные адреса релеев
  grep Received $INSPECT_DIR/in.$$ >> $INSPECT_DIR/vr1.$$
  echo "Mail from: $2 (may be forget)" >> $INSPECT_DIR/vr1.$$
  echo "To: $4" >> $INSPECT_DIR/vr1.$$
  grep Subject $INSPECT_DIR/in.$$ >> $INSPECT_DIR/vr1.$$
  echo >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  echo "* Virus(es)                                 *" >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  # Включаем в отчет список вирусов
  cat $INSPECT_DIR/vr.$$ >> $INSPECT_DIR/vr1.$$
  $SENDMAIL -f $VIRUSADMIN -r $VIRUSADMIN -F "Antivirus" $VIRUSADMIN < $INSPECT_DIR/vr1.$$
  exit 0
  ;;
*)
  # Произошла ошибка в работе антивируса. Сообщим об ошибке админу
  echo "Subject: ANTIVIRUS FAILED" >> $INSPECT_DIR/vr1.$$
  echo >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  echo "* Antivirus Failed with next problem        *" >> $INSPECT_DIR/vr1.$$
  echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  case "$AV_RESULT" in
    40)
       echo "* Unknown option passed.                    *" >> $INSPECT_DIR/vr1.$$
```

```
      ;;
   50)
      echo "* Database initialization error.              *" >> $INSPECT_DIR/vr1.$$
      ;;
   52)
      echo "* Not supported file type.                    *" >> $INSPECT_DIR/vr1.$$
      ;;
   53)
      echo "* Can't open directory.                       *" >> $INSPECT_DIR/vr1.$$
      ;;
   54)
      echo "* Can't open file. (ofm)                      *" >> $INSPECT_DIR/vr1.$$
      ;;
   55)
      echo "* Error reading file. (ofm)                   *" >> $INSPECT_DIR/vr1.$$
      ;;
   56)
      echo "* Can't stat input file / directory.          *" >> $INSPECT_DIR/vr1.$$
      ;;
   57)
      echo "* Can't get absolute path name of current     *" >> $INSPECT_DIR/vr1.$$
      echo "* working directory.                          *" >> $INSPECT_DIR/vr1.$$
      ;;
   58)
      echo "* I/O error, please check your filesystem.    *" >> $INSPECT_DIR/vr1.$$
      ;;
   59)
      echo "* Can't get information about current user    *" >> $INSPECT_DIR/vr1.$$
      echo "* from /etc/passwd.                           *" >> $INSPECT_DIR/vr1.$$
      ;;
   60)
      echo "* Can't  get  information  about  user        *" >> $INSPECT_DIR/vr1.$$
      echo "* clamav (default name) from /etc/passwd.     *" >> $INSPECT_DIR/vr1.$$
      ;;
   61)
      echo "* Can't fork.                                 *" >> $INSPECT_DIR/vr1.$$
      ;;
   63)
      echo "* Can't create temporary files/directories    *" >> $INSPECT_DIR/vr1.$$
      echo "* (check permissions).                        *" >> $INSPECT_DIR/vr1.$$
      ;;
   64)
      echo "* Can't write to temporary directory (please  *" >> $INSPECT_DIR/vr1.$$
      echo "* specify another one).                       *" >> $INSPECT_DIR/vr1.$$
      ;;
   70)
      echo "* Can't allocate and clear memory (calloc).   *" >> $INSPECT_DIR/vr1.$$
      ;;
   71)
      echo "* Can't allocate memory (malloc).             *" >> $INSPECT_DIR/vr1.$$
      ;;
   *)
      echo "Unknown error $AV_RESULT" >> $INSPECT_DIR/vr1.$$
      ;;
  esac
      echo "*********************************************" >> $INSPECT_DIR/vr1.$$
  $SENDMAIL -f $VIRADMIN -r $VIRADMIN -F "Antivirus" "$VIRADMIN" < $INSPECT_DIR/vr1.$$
  exit $EX_TEMPFAIL
  ;;
esac

exit 0
```

В данном скрипте реализовано уведомление только администратора (имея дело с современными почтовыми вирусами уведомление отправителя и получателя просто противопоказано), зараженные письма просто убиваются.

Перестартуем наши почтовые сервисы:

```
/etc/init.d/postfix restart
/etc/init.d/clamsmtp restart
/etc/init.d/clamav-daemon restart
```